(2)

AD A126945

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

AN INTEGRATED DESIGN SPECIFICATION FOR THE HARPOON SHIP-BOARD COMMAND-LAUNCH CONTROL SET (HSCLCS), AN/SWG-1A(V)

by

Orville Lawrence Sentman

and

Randall Keith Maroney

December 1982

Thesis Advisor: Ronald Modes

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A126945	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Integrated Design Specification for the HARPOON Shipboard Command-Launch Control Set (HSCLCS), AN/SWG-1A(V)		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis December 1982
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Orville Lawrence Sentman and Randall Keith Maroney		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		12. REPORT DATE December 1982
		13. NUMBER OF PAGES 135
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

HARPOON, Cruise Missile, Weapons System Design, Missile, Fire Control System

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Block 1C version of the surface launched HARPOON cruise missile has performance capabilities that cannot be used due to the limitations imposed by the HARPOON Shipboard Command-Launch Control Set (HSCLCS). This thesis is an initial effort to redesign the HSCLCS from the software engineering approach. The HSCLCS system specifications are derived from stated U.S. Navy's requirements and additional features proposed by the authors. The HSCLCS software design is completed in detail through the system definition. Development of the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

software design continues through the use of system data flow diagrams and their subsequent mapping into the preliminary system software structure. First iteration data structure definitions and functional module descriptions are provided.

Approved for public release, distribution unlimited

An Integrated Design Specification for the HARPOON Shipboard Command-Launch Control Set (HSCLCS), AN/SWG-1A(V)

by

Orville Lawrence Sentman
Lieutenant Commander, United States Navy
B.S.A.E., Purdue University, 1971

and

Randall Keith Maroney
Lieutenant, United States Navy
B.S., University of Texas, 1974

Submitted in partial fulfillment of the
requirements of the degree of

MASTER OF SCIENCE IN ENGINEERING SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 1982

Authors: ______________________

____

Approved by: ______________________
Thesis Advisor

____
Second Reader

____
Chairman, Department of Computer Science

____
Dean of Science and Engineering

ABSTRACT

The Block 1C version of the surface launched HARPOON cruise missile has performance capabilities that cannot be used due to the limitations imposed by the HARPOON Shipboard Command-Launch Control Set (HSCLCS). This thesis is an initial effort to redesign the HSCLCS from the software engineering approach. The HSCLCS system specifications are derived from stated U.S. Navy's requirements and additional features proposed by the authors. The HSCLCS software design is completed in detail through the system definition. Development of the software design continues through the use of system data flow diagrams and their subsequent mapping into the preliminary system software structure. First iteration data structure definitions and functional module descriptions are provided.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

The authors would like to thank the following people. Cdr. Jim Williamson, formerly the NAVSEA Surface Launch HARPOON Project Manager, for his aid in our decision to undertake the HSCLCS study and his continued support through its completion. Mr. Tom Erskines of NSWSES Port Heuneme, as our initial source of HSCLCS and HARPOON information. Mr. Lee Minin, the current NAVSEA Surface Launch HARPOON Project Manager, for allowing us to present our views of the HSCLCS Draft Specifications at the November, 1982 specification review. The professionals at McDonnell Douglas Astronautics, St. Louis, Mo., Mr. Fred Gais, Mr. Don Franke et. al., for their encouragement and interest in this report. Lieutenant Commanders Ron Modes and Bill Shockley for driving us to a timely completion of this report. Finally, and most importantly, to our wives, [redacted] [redacted] and our children for supporting the successful completion of our work.

I. INTRODUCTION

The introduction of successive block enhancements to the missile subsystem of the HARPOON Weapon System has rendered the existing HARPOON Shipboard Command-Launch Control Set (HSCLCS), AN/SWG-1A(V), inadequate to effectively control all the design features of the enhanced versions of the HARPOON cruise missile. Many of the features introduced by these block improvements cannot be controlled or exercised by the current HARPOON Weapon System operator. Futhermore, the anticipated block enhancements technically achievable in the remainder of this decade will certainly intensify this deficiency.

The HSCLCS must be redesigned to accommodate the diversity and minimize the burden on the operator of this increasingly complex system. The expansion of the operator interface to incorporate a graphic quality display and the use of fast and accurate manual input devices for system control will reduce the operator's burden. The design choice to use software programmable input keys increases the longevity of the new hardware suite.

The software engineering methodology chosen is that proposed in Reference 1. A representation of this approach is shown in Figure 1-1.

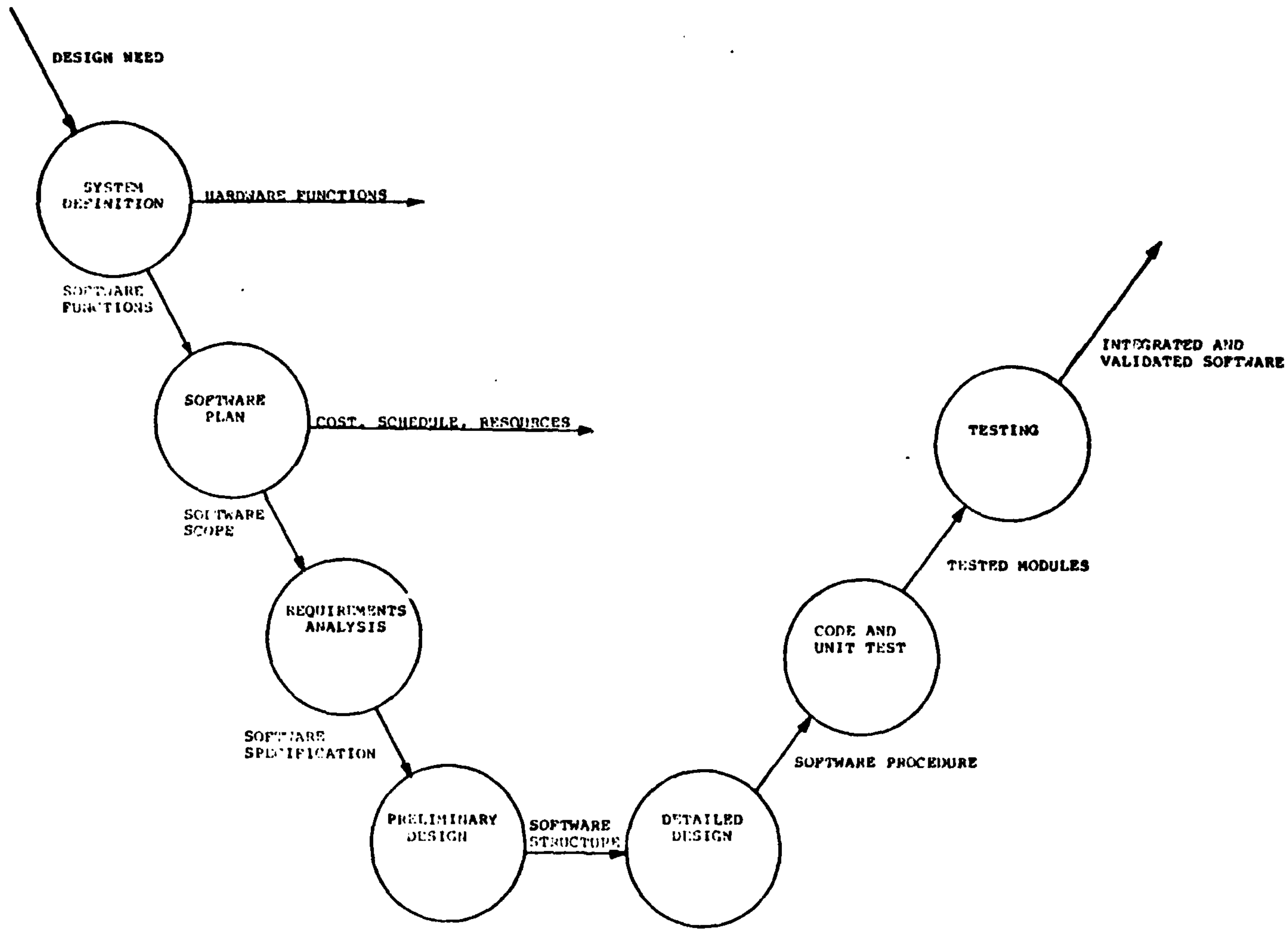

Figure 1-1 Representation of Software Engineering Procedure as Described in Reference 1

Follow-on work is essential to complete the HSCLCS software design beginning with a review of this report and continuing through the detailed design, coding, unit testing, and integrated system testing, represented in the last three circles of Figure 1-1.

II. SYSTEM SPECIFICATION

The purpose of this chapter is to specify the scope of the HARPOON Shipboard Command-Launch Control Set (HSCLCS), AN/SWG-1A(V), software and system development process. This chapter represents the first step in the planning phase of the software engineering process and achieves the following: summary of the existing system, statement of the needs intrinsic to the existing system, statement of the technical constraints imposed by hardware considerations, and identification of limitations levied by external factors. The product of this design step is the definition of the functional requirements for the HSCLCS.

A. EXISTING HARPOON WEAPON SYSTEM

The HARPOON Weapon System (HWS) has been developed by the U.S. Navy to fulfill the Navy's anti-ship mission requirements. The HWS is currently deployed on a wide assortment of ships, aircraft, and submarines to provide an all-weather, day or night, anti-ship capability at over the horizon ranges.

The HWS is comprised of the missile subsystem, the command and launch control subsystem, and associated launcher subsystems.

The HARPOON missile employs low level cruise trajectory, active radar guidance with counter-counter measures, and terminal maneuvering to attack surface targets at over the horizon ranges. The missile itself is essentially identical for ship, submarine, and aircraft launch. A booster is added for ship and submarine launch. The HARPOON is ship launched from a variety of launcher configurations.

The ship launched HARPOON missile employs either onboard or third party sensor data for targeting initialization. The operation is "launch and forget" since no ship control is required subsequent to launch.

For surface ship installations, the HWS control and data processing functions are provided by HSCLCS, which has three modes of operation: normal, casualty, and training. In the normal mode the major functions provided by the HSCLCS include:

- The distribution of power to various HWS equipment.
- The selection and application of missile warmup power.
- The ability to conduct various automatic and manually initiated tests which confirm the operability of the HSCLCS.
- The distribution of ship motion and speed data from ship equipment.
- The selection, transfer, processing, and display of target data.
- The coordination of the selection of the tactical missile mode and fusing.

- The selection of the launcher cell containing the intended HARPOON missile.
- The initialization of the selected missile and the supervision of the exchange of data between the missile and other HWS equipment.
- The control of all missile firing activities.

These functions are integrated and implemented to a large degree by the HARPOON Weapon Control Console (HWCC) and the Weapon Control Indicator Panel (WCIP). In most ship class configurations, the HWCC and WCIP are co-located in Combat Information Center (CIC).

The HWCC contains most of the HARPOON system-unique command and launch subsystem equipment, including the Data Processor Computer (DPC), the Data Conversion Unit (DCU) and HWCC life support equipment. Together these HWCC components perform data processing and conversion among various message types and provide interfacing with existing sensor and ship's equipment.

The WCIP provides visual status information to the operator during formulation of the fire control solution, and additionally provides manual controls for the operator.

The DPC is a 16 bit microcomputer with 15K of EPROM memory. The DPC utilizes an assembly language program to provide the following service functions:

- Launch envelope parameter validation.
- Missile command generation for implementation of missile control parameters including ship's attitude, search pattern orders, engine starting, flight termination

range, altimeter timing, and various selectable flight trajectory and maneuvering modes.

- Pre-launch testing of the missile.
- Pre-launch sequencing and timing.
- Data formating and transfer synchronization.

Additionally, various ship classes employ class unique component cards which are tailored to launcher interfacing and control for the respective launcher configuration.

The DCU processes all digital and analog signal conversions as required by installed hardware. The DCU in particular provides interfacing of target data inputs such as from the Naval Tactical Data System (NTDS) Slow Interface. Ship motion parameter data also is converted in the DCU.

1. Operational Targeting Sequence

Preparatory to a normal launch, the present HSCLCS is sent individual target data from an existing shipboard system. The HARPOON operator then selects one of three discrete seeker search patterns and then selects the seeker activation ranges and bearings as appropriate. Alternately, if the targeting range data is unknown, or known to be inaccurate, a bearing-only-launch (BOL) may be selected.

As a standby launch mode, the HARPOON operator can elect to manually input targeting range and bearing data into the HSCLCS. The source of this data may be shipboard systems or third party sources.

B. PROBLEMS ASSOCIATED WITH EXISTING HSCLCS

Successive block improvements have introduced added command and control complexity. With the present WCIP's buttons and display, the operator is ill-equipped to conceptualize, direct, and execute a well-formulated HARPOON attack in a timely manner. The missile capabilities, with continued block improvement, cannot become an operational reality without substantial hardware development and modification of the WCIP. Example of a typical WCIP is pictured in Figure 2-1.

The life cycle maintenance cost of the present HSCLCS will continue to be relatively high because existing software is written in assembly code and is heavily hardware dependent. Additionally, several different hardware configurations exist for subsurface, air, and surface launch of the common missile.

At the WCIP, the primary point of control of the HARPOON system, no tactical display is available for the operator to readily and directly evaluate the tactical surface situation.

With regards to HARPOON engagement planning, the following HSCLCS deficiencies exist:

- Full tactical control of existing missile variants (the pre-launch selectables) is precluded by the WCIP. Many of these variant features are inaccessible to the operator.

- The WCIP provides inadequate control for a well coordinated, multi-ship or multi-platform attack against a single surface target.

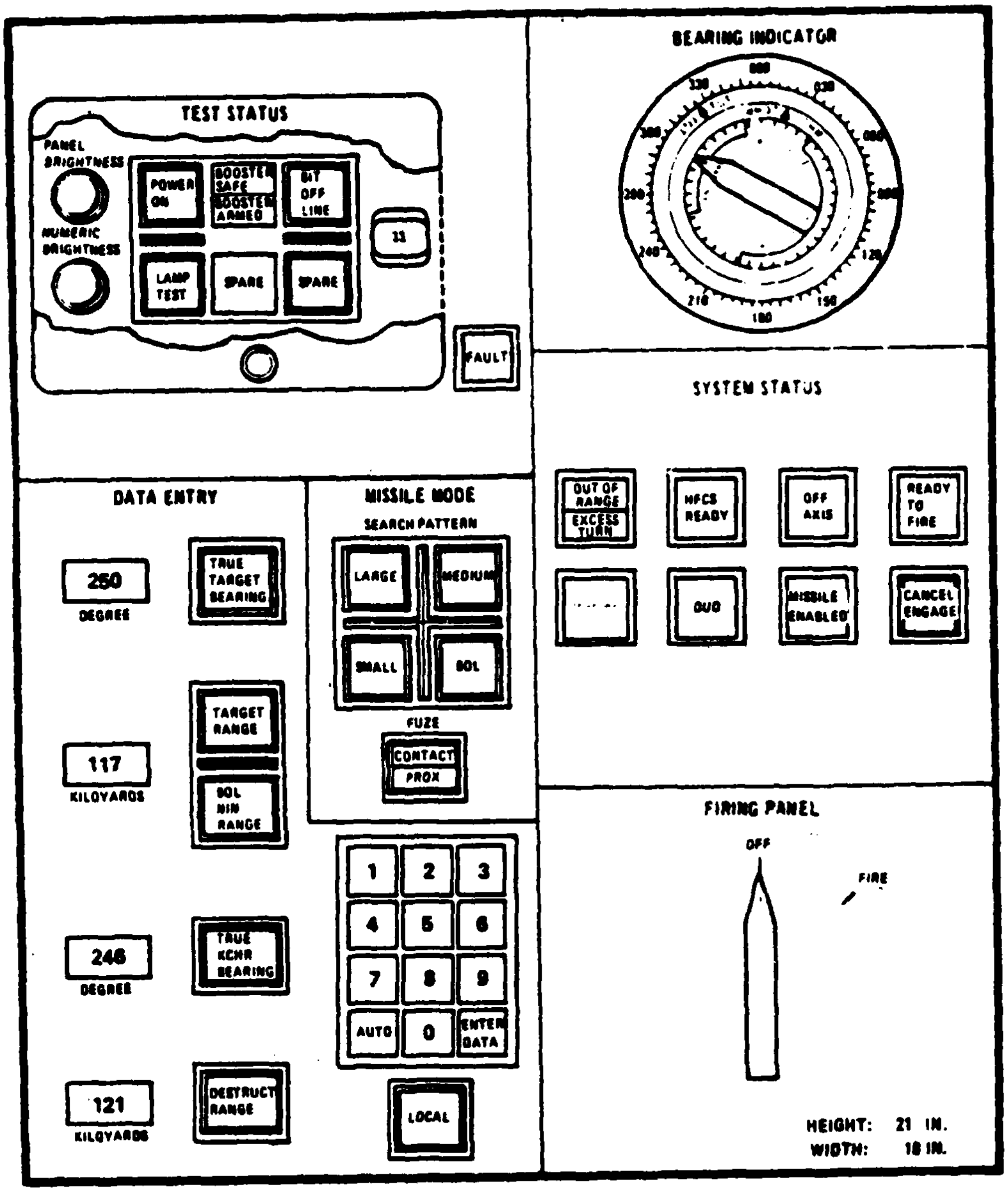

Figure 2-1 Representative Cannister Launch System WCIP

- The WCIP provides inadequate control for a multi-missile (i.e., "Ripple" or salvo) attack against a single surface target.

- The WCIP does not support the incorporation of available intelligence (e.g. target class, course, axis of vulnerability) into the engagement planning process.

- No computer-aided engagement planning is implemented.

With regards to the analysis of the engagement plan, the following HSCLCS deficiencies exist:

- Insufficient information is displayed at the the WCIP to permit the operator to evaluate the quality of an engagement plan (for instance probability of acquisition).

- Insufficient information is displayed at the WCIP to provide accurate indication of implied risk to unintended targets during booster drops, flyout and target acquisition.

- The WCIP provides no display of planned trajectory, flight path, or search patterns.

- The HSCLCS does not provide computer-aided engagement plan quality and safety analysis.

The WCIP provides no resource management status information on available missile and associated launcher.

The existing system has no means to accumulate or assimilate track data furnished by own ship's sensors or those from third party sources. Track data can be stored as only one track at a time with no provision for multi-track data retention.

Environmental parameters such as wind, rain, and sea state impact missile performance during flight. No means

exist to input or provide engagement corrections for these parameters.

Training in the normal at sea operating environment is of marginal quality due to lack of realism (i.e., no graphical engagement picture). Improvement in operator proficiency is therefore difficult to achieve.

C. HARPOON WEAPON SYSTEM CONSTRAINTS

The constraints defined in this section are principally technically oriented. Managerial constraints (e.g., cost, developmental procedures, etc.) are to be determined by competent authority at a later date.

Block 1A and Block 1B HARPOON missiles will continue to be operationally deployed throughout their normal service life. The upgrade for the HSCLCS must retain full launch operability with Block 1A, Block 1B, and Block 1C missiles.

The upgrade must be implemented such that the HSCLCS will continue to provide necessary prelaunch functions for transport, warm-up, aiming, and firing of all missile variants from all surface ship platform launcher configurations.

The upgrade must maintain interface compatability with the Naval Tactical Data System (NTDS) Slow Interface. Because of the total system loading already levied on NTDS, the upgrade HSCLCS is restricted from increasing the demands for input from NTDS. Any change in the data elements

required by HSCLCS from NTDS should be limited to data presently available in NTDS, or restricted to data used by other tactical systems. A reduction in NTDS processing requirements in support of HARPOON would be most desirable.

The presently existing launcher hardware configurations, including launcher control and test equipment, will not be subject to change for the upgrade HSCLCS.

The DCU hardware suite must remain intact. The DCU and DPC software is subject to change as necessary to implement the upgrade HSCLCS. Because of the assembly language implementation of present software, software change will be both difficult and expensive to develop, test, and maintain.

Due to acute shortages of space at HSCLCS operating stations onboard many surface ship platforms, the physical size of the HSCLCS is restricted to its present size.

The Built-In-Test (BIT) and Built-In-Test Equipment (BITE) requirements established for the existing HSCLCS will remain effective for the upgrade HSCLCS.

Overall system real time performance must meet operational requirements.

System reliability, hardware maintainability, and system environmental standards for the HSCLCS upgrade must meet or exceed the performance specified for the existing HSCLCS.

D. SYSTEM DEFINITION FOR HSCLCS UPGRADE

1. Introduction

The purpose of the system definition is to provide a comprehensive hardware and software description for the HSCLCS. This definition is subject to review and change by competent authority during the system development process as increased detail is acquired.

a. System Objectives

A statement of the HSCLCS upgrade objectives is considered essential for guidance in the system development process. The following discussion is a statement of these objectives developed by the authors.

The prime objective of the HSCLCS upgrade is the provision for full tactical deployment of all missile options incorporated in the Block 1C HARPOON missile design.

The complexity introduced by successive missile block enhancements has not been sufficiently addressed from the perspective of operator control. A substantial improvement in the degree of positive operator control during a tactical employment of the HWS is a system design objective.

To date, no graphical display representing the local tactical surface warfare scene has been directly available to the HARPOON operator. A tactical display will clearly improve operator comprehension of the tactical

situation during engagement planning and execution. This introduction of a tactical display and improved operator control mechanisms permits a more sophisticated employment of the HWS. The ability to plan multi-missile attacks and coordinated launches, from individual platforms against a common target, provides commanders greater command and control of anti-surface missile resources.

The existing HSCLCS is incapable of obtaining and retaining multiple targeting reports. A principal objective of the HSCLCS upgrade is the provision for the receipt, correlation, and display of multiple surface targeting data reports.

Historically, the introduction of successive HARPOON missile block enhancements has dictated WCIP hardware modifications. The use of programmable function keys for operator control alleviates this repetitive requirement.

Another objective for the HSCLCS upgrade is to provide the operator assistance in engagement planning and analysis. Automatic calculation and display of probabilities of acquisition is a valuable aid for the measurement of the relative quality of a planned engagement prior to the expenditure of missile resources. Autonomous generation of engagement plans for hostile tracks accelerates the plan formulation. The operator must retain positive control for the approval or modification of the plan.

b. System Operational Environment

Present internal shipboard warfare organization places the HARPOON operator under the supervisory control of the Ship's Weapons Coordinator (SWC). The SWC, or equivalent authority, orders the designation of a track as a target. The HARPOON operator, upon receipt of such orders, plans and executes the engagement. The typical HARPOON operator will be a senior enlisted or junior officer with proven experience and training in anti-surface warfare.

c. Hardware Component Overview

Figure 2-2 is a hardware component overview of the HWS. The only visible item of hardware change of the HSCLCS upgrade is the WCIP and its attached display console. Internal HSCLCS hardware changes are proposed for the DPC.

2. <u>System Hardware Functional Description and Allocation</u>

The HSCLCS upgrade requires neither functional nor physical changes in HWS hardware, other than HSCLCS subsystem hardware components.

a. HARPOON Missile Subsystem

The missile subsystem is not subject to change. All system functional specifications allocated to the missile subsystem of the existing HWS are germane.

b. Launcher Subsystem

The launcher subsystems of the HWS are not subject to change in the HSCLCS development process.

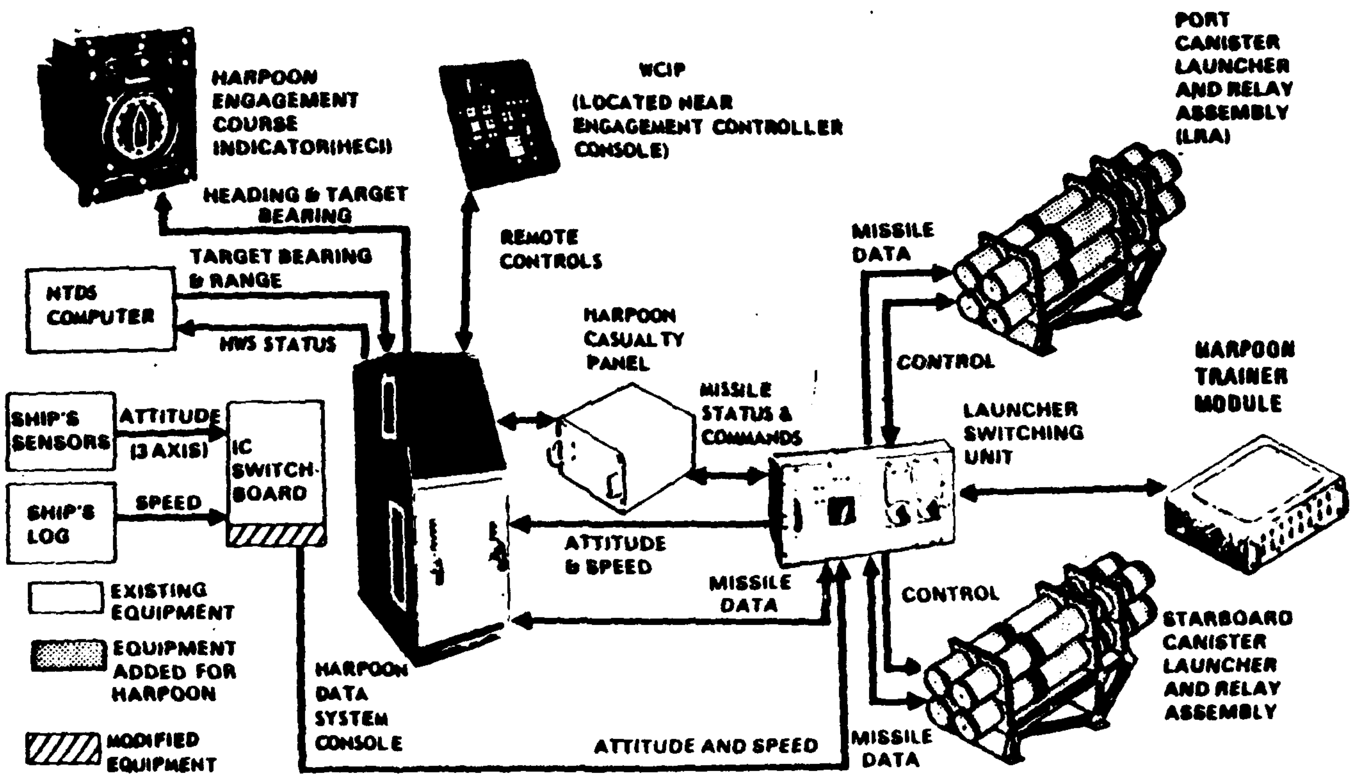

Figure 2-2 Hardware Component Overview of the Cannister Launch HARPOON Weapons System

Existing system specifications allocated to the individual launcher subsystem configurations remain effective and binding.

c. HSCLCS Subsystem

The HSCLCS upgrade requires extensive modification to HSCLCS hardware components. The functional specifications, currently allocated to existing HSCLCS hardware, remain virtually intact. However, additional functions are allocated for integration of improved hardware components.

(1) Weapons Control Indicator Panel. The most visible HSCLCS hardware alteration is the WCIP. A totally new display and operator console is planned for physical installation into the existing WCIP enclosure. Figure 2-3 pictures the preliminary design mockup of the replacement WCIP. Figure 2-4 is an enlarged view of the display area of the replacement WCIP and the associated function keys.

The new console provides a full tactical display and constitutes a major improvement for the HSCLCS upgrade. The proposed display is of a plasma media, vice the more standard cathode ray tube, and will provide a higher quality resolution of the display graphics. An attached microprocessor will process all screen graphics software routines as commanded by the DPC.

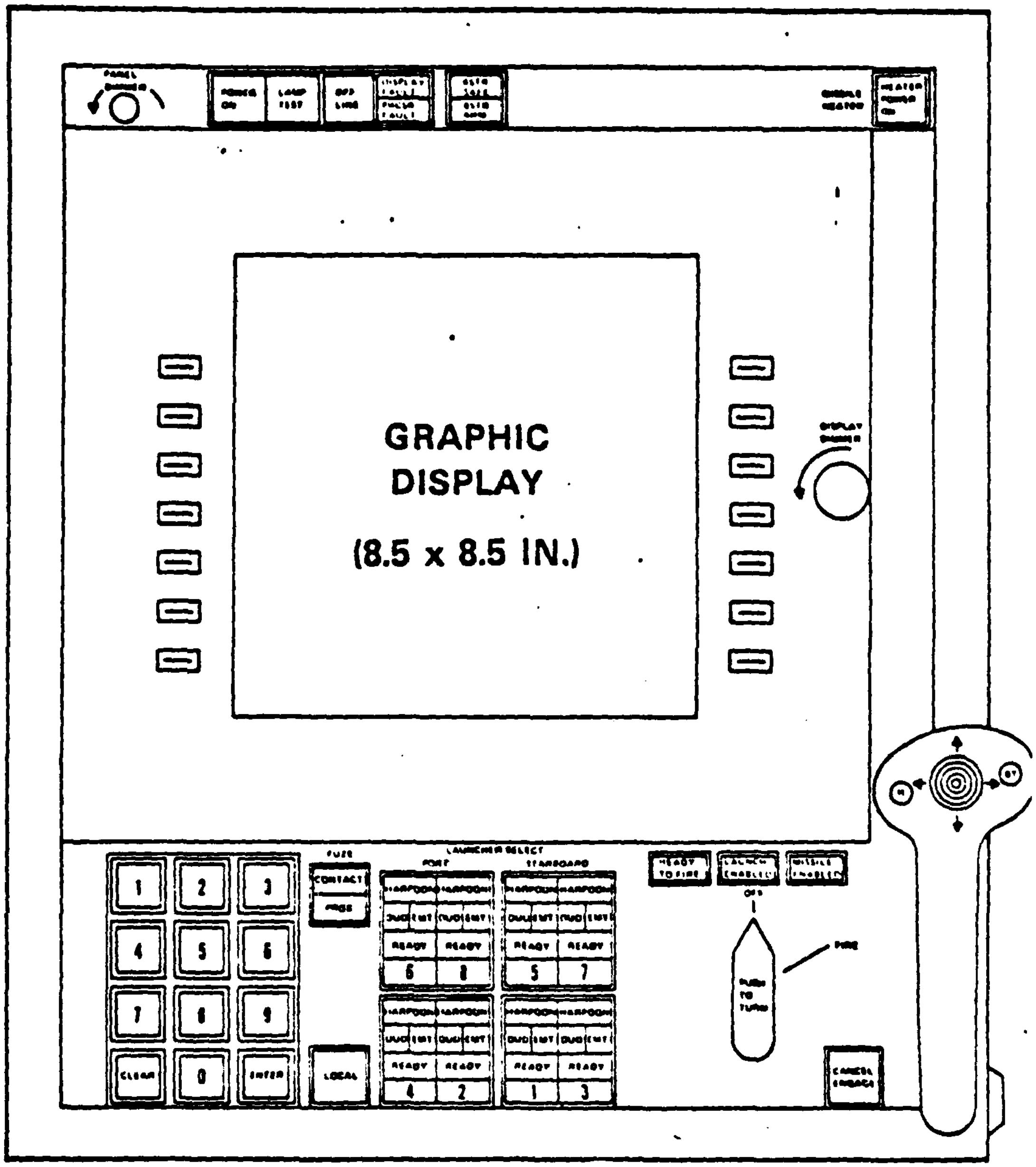

Figure 2-3 Proposed WCIP

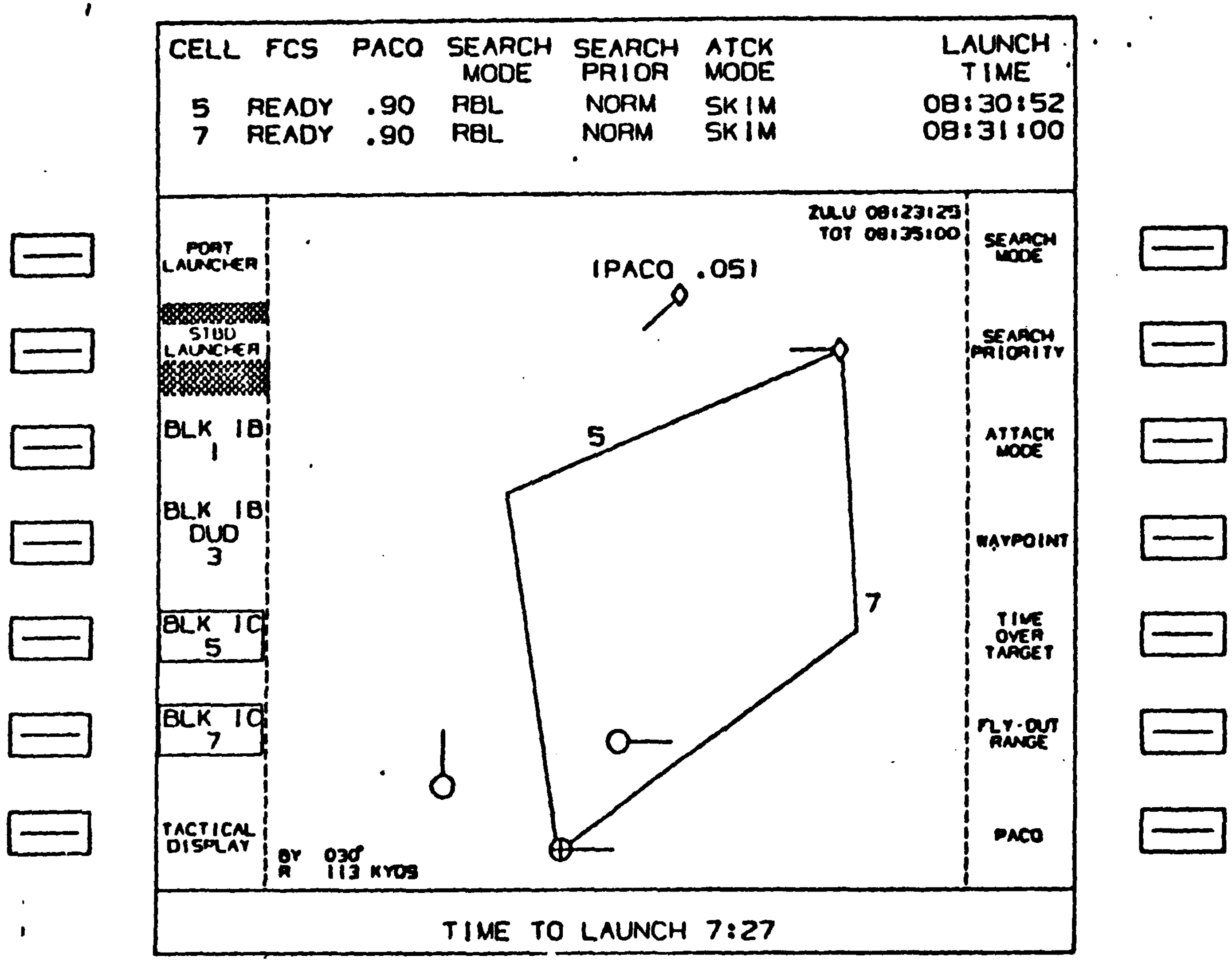

Figure 2-4 Proposed WCIP Plasma Display and Simulated Engagement

Co-located on either side of the display screen are a set of software programmable keys for use by the operator during HSCLCS operation. Software for the programmable keys resides in the DPC.

Cursor control is implemented by an isometric thumb button mounted on a stationary handle grip on the right side of the WCIP. Mounted on each side of the isometric cursor control button are a 'hook' actuating button and a 'break' actuating button. The 'hook' device signals to the HSCLCS software that the operator has positioned the cursor over a track or graphical position. The 'break' button nullifies a current 'hook' command.

Additional hardware associated with the WCIP include a firing key, a set of missile and launcher status indicator lights, and miscellaneous display console controls.

(2) Data Processing Computer. Hardware reconfiguration of the existing DPC cannot be determined at this stage of the HSCLCS development. Additional DPC memory is a minimum, established requirement for the HSCLCS upgrade. Preliminary research infers that replacement of the DPC microprocessor with a faster, militarized version of a commercially available CPU and additional memory is warranted. Note that existing functional specifications for DPC hardware remain effective. New software systems, with

the exception of display graphic software. will be processed by the new DPC computer system.

(3) Data Conversion Unit. Change to the existing DCU hardware is prohibited in the HSCLCS upgrade. DCU software changes are permissable only to the extent necessary to interface the data sources providing input for new DPC processing requirements. Any such change in DCU software functions shall be determined when full interfacing details are known.

(4) Display Processor. As previously mentioned, the display graphics software will be processed by an attached backend processor in the WCIP. Hardware functions allocated to this processor are:

- The acknowledgement of commands communicated by the controlling DPC.
- The decoding of DPC generated display commands.
- The generation of all display screen commands associated with the visual display.

3. System Software Functional Description and Allocation

All graphics display software is currently under development by the display console vendor and is not considered a part of this software specification. Graphics software processing is relegated to the display processor. As previously stated, data conversion software modifications requirements are unknown. All data conversion software shall continue to be processed exclusively by the DCU. Remaining

HSCLCS software shall be processed by the DPC. A detailed specification for this software follows.

a. General HSCLCS Software System Specifications

General purpose HSCLCS software includes essential interfacing, interprocess communications protocol, and state transition management.

(1) Interface Software Specifications. Detailed software requirements for interfacing various components of the HSCLCS are purposely deferred until sufficient hardware and interprocess communication details are resolved by follow-on research study.

(2) Software Support of Existing Missiles. HSCLCS software must maintain inter-operability with all USN missile subsystem variants through block 1C and RNSH missile variants. HSCLCS software must support the initialization, booster arming, and the Built-In Test (BIT) of all missile variants. Detailed software specifications of missile support functions are deferred until completion of follow-on research.

(3) Software Support of Existing Launchers. HSCLCS software is required to provide launcher support functions for all existing launcher configurations. Launcher support functions include at the minimum pre-launch transport, warm-up, aimpoint initialization, and firing sequencing for all missile variants through block 1C.

Additionally, launcher support software shall maintain compatibility with existing launcher BIT functions.

(4) Interrupt Handling Software. The software must provide for the acknowledgement of asynchronous interrupts from a variety of sources to communicate between devices, including:

- System launcher and missile monitors sending status reports.
- Operator control commands actuated by programmable control buttons on the WCIP.
- Operator cursor control inputs from the WCIP.
- Own ship motion sensor inputs including pitch, roll, speed through the water, course, and their respective time derivatives.
- Sensor targeting data reports from own ship sensors, NTDS, and from third party sources.

(5) CPU Resource Monitoring. System software must monitor CPU utilization. The scope of track data processing and engagement processing shall be variable as CPU resource availability becomes critical. This applies specifically to track history maintenance and scheduling of autonomous engagement plan optimization.

(6) Process Scheduling. The implementation of a concurrent, multi-programming embedded operating system is intended to provide greater flexibility for software system growth. Software functions are to be implemented by independent, concurrent processes. The operating system will of necessity perform the following minimum functions:

- Provide for the allocation of a CPU to ready processes.

- Implement synchronization primitives for each independent, concurrent process.

- Implement a priority system for the scheduling of ready processes. This priority assignment may be achieved implicitly through the use of a scheduling table.

(7) State Transition Management Software. The use of programmable function buttons for operator control of the system insures the long term functional utility of the WCIP display console. Human factors considerations dictate that the transition from one logical system control state to another be natural. With each state transition , the operator shall be given a new set of applicable choices of control options. To implement this in the system control architecture, the following are minimum software requirements:

- Provide display button labels for each operator control function. These labels must be organized into logical sets, or menus, which will be displayed as a unit. The menus correspond, one-to-one, with a given overall system control state.

- Implement a state transition matrix to provide a mapping from a given state to a corresponding menu, with individual button meaning uniquely defined for a given state.

- Provide for the sequencing of events necessary to implement a state transition. When a control command is received, decode the command. If a state transition command is invoked, the change in control state shall be recorded and a new screen menu sent to the display screen. A critical period, when no commands are to be read, exists during the actual transition sequence.

- Provide for the decoding of all state dependent inputs.

b. Operator Control Interface Software

The WCIP is the central point of control for the system. The plasma display and indicator lights are output from the HSCLCS to the HARPOON operator. Additionally, the WCIP provides the operator mechanisms for input of control commands and data. For the specification of software requirements these inputs and outputs, although physically furnished at the WCIP, are treated separately to minimize confusion and to improve clarity.

(1) Display Output Software. The HSCLCS control related display functions are as follows:

- Display programmable button labels indicating HSCLCS operator functional choices in a specifically reserved screen area adjacent to to the corresponding function button.

- Provide for the operator queues and messages in a specifically reserved screen area.

- Display illegal action alerts.

- Display a notice of lockout of any operator selected action.

- Display ZULU or local time as selected by the operator. The default time is ZULU.

- Display cursor position by a uniquely distinguishable symbol such as a small circle similar to NTDS cursor display.

Display is the only form of feedback to the operator during the engagement planning process. The display of options available and those selected provide problem solving continuity to the operator. The graphic

representation of the planned flight path permits rapid plan formulation and conceptual validation. Engagement related display functions are as follows:

- Display in a specifically reserved screen area, options as selected for each engagement plan.
- Display projected flight path for a planned or partially planned engagement.
- Display projected missile flight paths for missiles in flight subsequent to launch.
- Display time for launch for a planned attack.
- Display projected time of impact for a missile in flight.
- Display time desired for impact when a coordinated, multi-platform attack is selected.
- Display a data age alert for engagements using targeting data exceeding maximum age limitations.
- Display launch inhibit notices and the respective cause.
- Display a notice that the flight path, as planned, exceeds the maximum range of the missile variant selected.
- Display environmental parameters as they are set by the operator or as requested.
- Display cartographic land mass representation as entered by the operator.
- Display the booster drop zone projected for a given engagement plan.
- Display a graphic representation of waypoints when selected for an engagement.
- Display minimum and destruct ranges when selected for an engagement.
- Display a graphic representation of search pattern expansion when selected for an engagement.

- Display a graphic representation of active radar seeker search area for an engagement.

- Display the point of descent with a graphically distinguishable marker when high altitude fly-out is selected.

- Display the off-axis turn angle numerically in degrees for a selected aimpoint.

- Display the selected terminal attack mode.

- Display automatically the calculated probability of acquisition for an intended target.

- Display probability of acquisitions for unintended tracks as requested by the operator or if the calculated value exceeds an established maximum allowable threshold for unintended tracks.

- Display a graphic scale representation of the targeting uncertainty ellipse for the intended target of an engagement.

- Display a graphic scale representation of the targeting uncertainty ellipse for local unintended tracks as requested by the operator or under other conditions to be determined.

- Display missile ready notices.

- Display missile launch progress reports including cell or rail empty or missile dud.

- Display missile resource data including variant identifier, individual missile status (if other than fully operational), and cell or launcher location.

Track related display software functions are central to the declared objectives for improved tactical comprehension by the operator. The tactical display range scale along with the display point of reference determine the scope of the tactical area to be displayed. To reduce redundancy in the software requirements specification,

exceptions related to the scope of tactical display area are omitted. Track related display software functions are as follows:

- Display own ship's position with a graphically distinguishable symbol.
- Display surface tracks with the appropriate standard NTDS symbology.
- Display air tracks with the appropriate standard NTDS symbology.
- Display true course leaders of fixed length for surface contacts with a known course.
- Graphically distinguish an operator designated target.
- Display true bearing and range to a designated target.
- Display a notice of failure to correlate targeting data. And display the converted targeting data upon operator request.
- Display a graphically distinguishable line of bearing as input manually by the operator.

The display requirements associated with the Built-In Test system are applicable when the system is in the test mode only and are as follows:

- Display missile and HSCLCS BIT test results including "go or no-go" notice, failure status code when a failure is detected, and BIT 'no-go' evaluation reports.

(2) Manual Input of Commands and Data. As stated previously, the dual role of the WCIP as both an operator input station and an output device can be confusing. This section deals exclusively with software functional requirements for manual operator input. Software supporting

the following manual inputs is required. Manual operator inputs for control of the HSCLCS mode are as follows:

- Input to set the HSCLCS mode to the training mode.
- Input to set the HSCLCS mode to the test mode.
- Input to set the HSCLCS mode to tactical operation.

Several display operator control functions can be conceived for improvement of the overall flexibility of the display system. The operator may desire to temporarily suspend display a given class of tracks or may desire to turn off the track course leaders. The options are endless and resolution of which to implement is best deferred. The minimum set of manual operator display control functions are as follows:

- Input to set ZULU or local time.
- Input to set the display frame of reference to own ship's position.
- Input to set the display frame of reference to a geographical reference point.
- Input to set or change the range scale.

Manual operator input functions related to track data and track data maintenance are as follows:

- Input to set or change NTDS grid reference coordinates.
- Input track initialization targeting data.
- Input a track deletion command for an operator designated track.
- Input for a designated track position update data.
- Input for a designated track course update data.

- Input for a designated track speed update data.
- Input for a designated track size data.
- Input for a designated track identification data (e.g. DDG-2 or Kresta).
- Input for a designated track classification (i.e. friendly, hostile, or unknown).
- Input for a designated track platform type (i.e. air or surface, default to surface).
- Input a line of bearing for targeting. Line of bearing input shall consist of a true bearing from an operator designated position or track.

The rapid, accurate input of operator engagement data and the positive selection of tactical options is a primary objective of the HSCLCS upgrade. Engagement-related manual command and data input functions are as follows:

- Input a track designation command (i.e. "hook" a displayed track by placing the cursor near the desired track and pressing the "hook" button). Note that only one track can "hooked" at a time. Otherwise confusion would inevitably result when two or more tracks are in close proximity to a "hook".
- Input data to set or change environmental parameters including relative wind speed and direction, precipitation (i.e. 'yes' or 'no' with a 'no' default value), temperature in degrees Celsius (default to ambient).
- Input cartographic land mass plotting data comprised of true bearing and range "hooks" from the display reference point to prominent geographical land mass features.
- Input an override for any engagement plan selectable.
- Input an inhibit launch command.

- Input engagement plan confirmation (signaling operator concurrence and intent to proceed with the engagement sequence).

- Input a command to display probability of acquisition of an operator specified unintended track.

- Input seeker search pattern sizing parameters.

- Input waypoints for an engagement by 'hooking' points desired as flight trajectory waypoints.

- Input for the selection of the terminal mode (sea skim or pop-up).

- Input minimum and destruct ranges.

- Input for the high altitude fly-out mode and the associated range.

- Input search pattern expansion direction. Valid values are 'right', 'left', 'far', 'near', and the default value of 'normal'.

- Selection of the multi-missile attack mode against a designated target. Associated inputs are salvo size, intended time of arrival on target, and true target approach bearing.

- Selection of the coordinated, multi-platform attack mode. Associated inputs are salvo size, intended time of arrival on target, and true target approach bearing.

c. Track Data Base Maintenance System

The Track Data Base Maintenance System (TDBMS) software provides all track data processing for the HSCLCS. The software must permit the receipt of targeting data from diverse sources including manual input, NTDS, own ship's sensors, and third party sensors. This raw targeting data must be converted into a common reference system for correlation. The track data base system utilizes correlated data to maintain a current data base representing the

tactical surface scene for reference by various other software subsystems including the display system, the engagement planning system, and the engagement analysis system.

To reduce redundancy, reference to the host of targeting sources are omitted in the track data base system software requirements specifications.

(1) Input Source Compatability. The TDBMS software must preserve input source compatability. The minimum targeting data interfacing software functions are as follows:

- Asynchronously receive real-time targeting data from all sources.
- Convert all targeting data into the track data base reference coordinate system. The TDBMS reference coordinate system should be chosen so that overall processing requirements, for the track data base system and the software users of the track data base, are minimized.

(2) Track Data Base Maintenance Software Functions. The following are data base maintenance related functions associated with the Track Data Base Maintenance System:

- TDBMS software must provide for the initialization of a track record for both surface and air contacts as required by explicit 'new track' notification.
- TDBMS software shall maintain the own ship track record based on dead reckoning of ship's position and motion data.
- TDBMS software shall remove a track designated for deletion from the data base.

- TDBMS software shall be capable of deleting a designated track from data maintenance as explicitly commanded by the operator. Likewise, the software shall be capable of restoring a track, previously suspended from track maintenance, back to full track maintenance status.

- TDBMS software shall correlate incoming track data with existing track records and in the event of non-correlation implement a policy to be determined.

- TDBMS software shall provide for rapid access to an existing track by any user of the track data. If no such track exists searching is to be minimized. Access parameters should include position, a specified geographical area descriptor, classification, or track identifier. Suggested methods of implementing an access mechanism are a dense index of key values or multiple hashing functions. The hashing function indicates the better relative performance in the critical areas of access speed, versatility in accessing parameters, and maintenance of the accessing mechanism in the real-time.

- TDBMS software shall maintain an available track record pool to preclude memory allocation for new records at run-time and to improve data base reliability.

- TDBMS software shall, in the event of non-availability of allocatable track node resources, implement a policy to be determined.

- Write access protection by mutual exclusion of competing processes shall be provided on the track record level. Processes actually writing to a record are in fact in a critical region and must not be interrupted. Additionally, write access is restricted to TDBMS software only.

- Read access to a track record is unrestricted.

- Track records shall contain, at the minimum, the track position in the normalized track coordinates, track unique identifier, sensor source type source identifier, track size (small or large), targeting data quality indicator value, track history headed by last known course and speed, time stamp indicating the time of the most recent report, track classification identifier (i.e. hostile, friendly, or unknown), absolute track identifier by ship class or unit name, true bearing and range from own ship, a time stamp and a linkage pointer

to an established engagement plan where one exists for a particular track.

(3) Data Manipulation Functions. TDBMS software shall conduct the update of track positional and motion data based upon correlated, converted targeting data reports.

TDBMS software shall be capable of triangulating a track position from three or more specified lines of bearing.

TDBMS software shall, after a set elapsed period without additional targeting data for an established track, dead reckon the track position based on respective track and own ship motion data.

TDBMS software shall time stamp time relevant track data base processing results.

During periods of near capacity memory and CPU resource utilization, TDBMS software shall provide graceful degradation of non-essential track related processing (e.g. suspension of track history maintenance and superfluous track positional updates).

d. Engagement Planning System Software

The Engagement Planning System (EPS) is a software system whose purpose is to coordinate the formulation of an engagement plan for a designated target. The EPS shall routinely conduct autonomous engagement planning for known hostile tracks as CPU resources are available. Upon designation of a given target for

engagement, the EPS is under exclusive control of the operator. The EPS shall have access to the Track Data Base, the Missile Resource Data Base, own ship's parameter data, and the environmental data as required in the engagement planning process.

(1) General Engagement Planning Software Functions. The following are general EPS software functions which are applicable to all engagement modes:

- EPS software shall support engagement planning for all missile variants through Block 1C.
- EPS software shall respond to all manual and NTDS engagement related orders.
- EPS software shall provide for the control and use of all tactical missile selectables.
- EPS software shall be capable of computing the projected time of occurance of key events of a planned engagement. Examples of such events include the time of impact, time of arrival on target, time for seeker activation, time for waypoint course manuever, and time for launch to achieve a specified arrival time.
- EPS software shall be capable of calculating missile attack boundaries of an engagement plan and storing these attack boundary parameters in a format compatible with the format required by the display system.
- EPS software shall provide engagement planning for a concurrent, multi-missile attack with time of arrival as a determinant parameter.
- EPS software shall provide engagement planning for a coordinated, multi-platform attack for near simultaneous time of arrival on target.
- EPS software shall be capable of reading specific and non-specific track records from the Track Data Base. Access by a track unique identifier should return a specific record of track data associated with a particular target. Access by a category for track record

qualification should return a set of track records which each satisfy the categorical qualifier. Examples of queries of this type are : 'all hostile tracks' or 'all tracks with a range less than 100 miles and a true bearing from own ship of more than 25 degrees and less than 65 degrees.'

- EPS software shall be capable of reading specific and non-specific resource data from the Missile Resource Data Base.
- EPS software shall be capable of reading environmental data.
- EPS software shall be capable of reading ship's motion data from ship's motion parameter variables.
- EPS software shall record a finalized engagement plan in the Engagement Plan Data Base. Such a plan will record all tactical selectables, associated parameters, missile resources selected, identification of the designated target, and the time of plan formulation.
- EPS software shall provide manual override for any portion of an autonomous engagement plan. The operator may elect to substitute legal parameters for the parameters individually rejected or another entire plan.
- EPS software shall calculate the projected flight trajectory for a planned engagement and represent the trajectory in a display compatible format.
- EPS software shall submit a completed engagement plan to the Engagement Analysis System for engagement analysis.

(2) <u>Manual Engagement Planning Software Functions</u>. When in the engagement mode, EPS software shall provide for the logical and orderly selection of all missile employment options. As tactical variables are selected, they are properly recorded and displayed. A given selection may in turn determine another set of logical options to be presented to the operator. Options which are not applicable

for a particular operator selected missile variant shall not be presented to the operator.

The following manual engagement EPS software requirements are ordered in a sugguested, but non-binding, sequence for presentation to the operator during engagement plan formulation:

- Entry into the engagement mode shall be automatic upon operator designation of a target. Validation of the designated track as an engageable, non-friendly shall be implicit by the change in the displayed target symbology to that of an engaged target and the continuance of the engagement sequence.

- When an autonomously generated engagement plan exists for the designated target, the autonomous generated engagement plan shall be displayed by a graphic representation of the flight trajectory. A textual summary of the plan shall be displayed in the specifically reserved area along with the associated probability of target acquisition of the plan and the time of plan formulation.

- Software shall provide the operator the opportunity to approve all, or any portion of the plan. Operator changes shall be immediately displayed and recorded. If no such autonomously generated plan exists, the EPS mode will default to manual engagement and the operator shall be provided the options for formulating a complete engagement plan.

- The operator shall be able to select the missile or missiles from the available missile resource inventory for execution of the engagement.

- The operator shall be able to select any logical tactical missile option that is supported by the missile variant selected and that is consistent with options already selected in the engagement formulation sequence.

(3) <u>Automatic Engagement Planning Software Functions</u>. The provision for autonomous engagement planning support functions is an objective for the HSCLCS upgrade.

Concrete performance criteria for the autonomous plan generating software remain to be established as feasibility is better defined.

As CPU resources become idle, conduct autonomous engagement planning of hostile tracks. Note, this is the lowest priority of any CPU scheduling requirement. An intuitive strategy is to first conduct trial and error on a limited set of simple and effective engagement plans (such as a straight line of sight launch). If no simple solution is found which is both safe and effective in terms of probability of target acquisition, then an iterative process to find a plan would be conducted.

At the minimum, autonomous engagement planning software shall be capable of selecting the missile terminal mode based on known target size, selection of the fly-out mode, selection range and altitude required to clear shipping obstructions, and the selection for launch the missile variant with the least performance options which is still capable of executing the engagement plan. Autonomous engagement planning software functions shall include automatic waypoint selection to preclude a high probability of acquisition for unintended targets during autonomous engagement planning. Additionally, the autonomous engagement software shall directly support manual engagement. Provide for automatic waypoint selection to insure simultaneous

arrival on target of the salvo missiles when multi-missile mode is selected.

e. Engagement Plan Analysis Software Functions

The analysis of engagement plans is a stated objective of the HSCLCS upgrade. Each planned engagement shall be submitted to the Engagement Analysis System (EAS) for evaluation prior to execution of the plan by a missile launch. Engagement plan analysis is concentrated on the relative quality of the engagement plan as measured by the probability of acquisition and the relative safety of the plan in terms of the threat posed by its execution to unintended targets and flight path obstructions.

(1) Engagement Plan Quality Evaluation Software Functions. EAS plan software shall be able to calculate the probability of acquisition for any track in the track data base. The probability of acquisition for the intended target shall be calculated for each plan. The probability of acquisition of unintended tracks shall be calculated on a selective basis dependent upon the proximity of the track to the projected search area.

Flight path length shall be evaluated to permit operator notification of engagement plans with a projected flight path in excess of the range performance maximum of the missile variant selected.

(2) Engagement Plan Safety Confirmation Software Each engagement plan submitted to the EAS software for evaluation shall be examined to insure that the planned flight path does not intersect at a sea skim altitude the targeting uncertainty ellipse of an unintended track.

EAS software shall ensure that projected booster drop zones do not threaten friendly surface tracks.

f. Resource Management Functional Software

The Resource Management System (RMS) maintains data on missile and launcher subsystems. This data is used by the EPS in both the manual and autonomous modes to validate the availability and operability of the missile and launcher pair. The intelligent management of missile and launcher resources is a stated objective for the HSCLCS upgrade.

The RMS software shall provide for receipt, conversion, and storage of missile and launcher resource data. This data includes missile status reports, launcher status reports, and missile loadout data. Additionally, supplementary data may be manually input by the operator.

The RMS shall maintain a special purpose data base with current data on missile inventory and status by each individual launcher. RMS software shall support limited queries on the RMS data base.

III. SOFTWARE PLAN

The software plan is the second major step in the planning phase of software engineering. The derivation of the software plan combines two tasks: research and estimation [Ref. 1]. Research determine the scope of the software module in the software design. Estimation is performed through the detailed evaluation of the functional description of the System Specification from Chapter II to guide cost and feasibility assessment of the system design.

> "The objective of software planning is to provide a framework that enables the manager to make reasonable estimates of resources, cost, and schedule. These estimates are made within a limited time frame at the beginning of the software project."[Ref. 1]

A. HSCLCS SOFTWARE PLAN

The software plan is developed where costs, manpower, and scheduling are important and difficult to plan such as in industry. The software specification will be used as the software plan for purposes of continuity of the software engineering design method shown in Figure 1-1. The authors strongly recommend that a HSCLCS project software plan be tailored to the costing, manpower, and scheduling requirements peculiar to the graduate academic environment of the Naval Postgraduate School. Such a plan is considered

essential to the ultimate successful completion of the project.

B. BASIC REQUIREMENTS OF THE SOFTWARE PLAN

The basic requirements of the software plan are listed here as a reference guideline.

- Keep the software plan physically short.
- Establish validity of the software document.
- Describe what the software is.
- Describe the cost of the software.
- Describe the length of the software development.

C. SOFTWARE PLAN STRUCTURE

A skeleton of the basic software plan is shown here [Ref. 1].

```
1.0  Scope
     1.1  Project Objectives
     1.2  Major Functions
     1.3  Other Characteristics
     1.4  A Developmental Scenario
2.0  Resources
     2.1  Human Resources
     2.2  Hardware Resources
     2.3  Software Resources
     2.4  Availability Window
3.0  Cost
4.0  Schedule
```

IV. SOFTWARE REQUIREMENTS ANALYSIS

Requirements analysis is the first step in the planning phase of the software engineering development. The purpose of this step is to fulfill the following objectives [Ref. 1]:

- Provide a foundation for the software development by uncovering the flow and structure of information.
- Describe the software by identifying interface details, providing an in-depth description of functions, determining design constraints, and defining software validation requirements.
- Establish and maintain communication with the user-requester and the developer so that the preceeding two objectives may be satisfied..

A. AREAS OF REQUIREMENTS ANALYSIS

Software requirements analysis is divided into four areas [Ref. 1]:

1. Problem Recognition

Problem recognition requires review of the software specification and the software plan. For the interim, software specifications will be used as the software plan. The analyst must not only review these plans, but must form communications links to the project manager (overall project coordinator), user/requester, and software team.

2. Evaluation and Synthesis

Evaluation and synthesis is the major effort during the software requirements phase. The flow of data and its structure, detailed refinement of the software functions, and

discovery of design constraints are the steps to accomplish this portion of the design process.

3. Software Requirements Specification

The software specification is the deliverable derived from the two steps above. The Specification provides the requester with the design that shows the development of the software from II. SYSTEM SPECIFICATION and III. SOFTWARE PLAN.

4. Software Requirements Specification Review

The software specification is thoroughly reviewed by the user, requester, and developer to ensure that the specification properly reflects the needs of the user/requester and that the design is feasible from the software engineering viewpoint.

B. DATA FLOW DIAGRAM (DFD)

The data flow diagram (DFD) is a graphical aid for depicting the data flow of the software system being designed. A complete understanding of the DFD is imperative to the understanding of the software engineering design method adopted in this paper. The following is a synopsis of the use of the DFD:

1. DFD Attributes

- Information flow in any system can be represented by a DFD.
- Each "bubble" or transformation in any DFD may require significant refinement to establish complete understanding.

- Emphasize data flow. Do not worry about control of the data.

2. DFD Symbols

- Information (i.e., data flow) is represented by a labeled straight line from the source to the sink with the arrowhead pointing to the sink.

- A process data transformation is represented by a circle called a "bubble" with a meaningful label.

- Information sources and sinks are represented by rectangles with a meaningful label.

- Stored information (e.g., data bases or files) are represented by two lines in parallel with a meaningful label.

3. DFD Usage Guidelines

- The first layer of the DFD is always the system module.

- The second layer of the DFD should be the generalized or 'overview' DFD.

- All arrows, bubbles, sources, sinks, etc. must have meaningful names as labels. Label specifically all transforms (bubbles) with a transitive verb to denote the action of the transform, and with a nonplural object to complete the description.

- Information continuity is required on DFD refinements. All incoming and outgoing arrows in the DFD being refined must appear in the refinement.

- Refine only one bubble at a time. The bubbles in the overview DFD are numbered with a single integer beginning at '1'. Then as they are subsequently refined, the expansion's numbers are added to by a '.' and another integer beginning at '1'. This numbering system is continued for all DFD's. As an example, if the overview has three transforms, they are numbered 1, 2, and 3. Then for the refinement of 1 into three new transforms the resultant transforms numbering is 1.1, 1.2, 1.3. Further expansion of 1.1 to two new transformations, would be numbered 1.1.1, 1.1.2. Recall these transforms all require an appropriate name as described above.

- Bubble refinement can yield bubbles, rectangles, or two lines in parallel in any combination. (See DFD signals IV.B.2).

- DFD's allow isolation of any domain of change.

- When there is uncertainty whether the DFD development is complete, assume that further refinement is possible and continue with the DFD refinement process.

- Follow data flow as a single thread from left to right. The DFD development may require a loop back to a previously defined transformation. Provisions for the single thread data flow where such a loop is required are made by duplicating the transformation so the flow continues from left to right.

- A transformation may output control data for a subordinate module. This control data does not represent control structure and therefore is not control flow.

C. HSCLCS DATA FLOW DIAGRAMS

Figures 4.1 thru 4.10 represent the development of the HSCLCS system by the data flow method described in section IV.B.

The fundamental HSCLCS DFD is depicted in Figure 4-1. The HSCLCS bubble is the domain of change that will be developed in the subsequent DFD's. The transformation labeled NTDS indicates a domain of change also. The development of this NTDS transformation is not pursued herein.

The second layer (first refinement) DFD of the HSCLCS is shown in Figure 4-2. The refinement of this DFD from Figure 4-1 is dramatic. Five new bubbles are now available for expansion in future refinements. These five bubbles are derived from the data flow analysis and are numbered to aid

reference to subsequent refinements. These transforms constitute the heart of the new HSCLCS and have the following basic function:

- Transform 1 - **Process Input** receives and processes all manual inputs and transforms the data so that it may be properly routed to one of the other four transforms. Note that this transform represents the input side of the WCIP, while transform **4 - Decode Output** represents the transformation of the outputs to the screen display, and all the other visual displays that are a part of the new WCIP.

- Transform 2 - **Update Track Data Base** processes both the manual input of **1 - Process Input** and the NTDS track data input and maintains a track data base for use by **4 - Decode Engagement** and **5 - Plan Engagement** transforms.

- Transform 3 - **Convert Environmental Data** provides a means for the operator environmental data input to be stored in a data base for use by **4 - Decode Output** and **5 - Plan Engagement** for display and engagement purposes respectively.

- Transform **4 - Decode Output** takes all the data from the databases maintained by the various transforms and operator manual display orders then provides the transformation required for proper display.

- Transform 5 - **Plan Engagement** develops and sends launcher/missile orders when it receives the orders from the operator thru **1 - Process Input.** Perhaps the most complex algorithm is also contained in this transform, that of determining an automatic engagement solution to some degree of optimization. One simple engagement algorithm may calculate only straight shots at the target. Complexities arise if waypoints are required, and even more complexities if waypoints are determined automatically by this transform.

The complete development of the DFD for the **1 - Process Input** transform is shown in Figure 4-3. The operator inputs his manual input order. This bubble identifies the transaction (e.g. signal, event, or unit of data that

triggers or initiates some action) and passes the data required to the proper receiving transform. The control mechanism for this is not considered in DFD development.

Figure 4-4 is the first refinement of the track data base DFD and leads to six new transforms.

Figure 4-5 is the complete refinement of the data base DFD. Four additional transforms are derived. Note that tracks may be deleted by a user manual input function or by NTDS.

The complete transform of 3 - **Convert Environmental Data** is shown in Figure 4-6. This is a simple transform that enters environmental data into the environmental data base and time stamps the data.

Expansion of the **4 - Decode Output** leads to eight new transforms depicted in Figure 4-7. Transform **4.1 - Select Function** is driven by the manual display order. For the purposes of the DFD development, any of the data flows are possible, so each data flow from this transform will result in a display order through the intermediary transform. Transforms **4.2** thru **4.6** all send display orders to transform **4.7 Console Display A/I** which insures a proper interface with the WCIP. Note the tracks from the track data base are transformed via **4.8 Convert Track to Screen Coordinates** and that this function is not manually ordered.

5 - Plan Engagement transform is the most detailed DFD development. Figure 4-8 shows the first refinement.

Transform 5.1 and 5.3 provide interfaces to databases for ship parameter data and launcher missile status respectively. This data is time stamped so that its relative age can be judged by those modules which use this information. **5.2 Plan Engagement** plans engagements continually when the CPU is available for its use. This attempts to keep the CPU busy and not in an idle state. **5.4 - Assign Launcher Missile Order** requests an engagement solution of **5.2 Plan Engagement** when operator orders, assigns launcher, missile, and provides launch order to missile. The missile is launched by a separate function on WCIP.

Figure 4-9 is a refinement of 5 - **Plan Engagement.** Transforms 5.2 and 5.4 are the only modules of Figure 4-8 which require further refinement. Transforms **5.2.1** through **5.2.4** take track data, calculate the uncertainty ellipse, and acquisition and compare to any existing solution existing in the engagement track data base to determine if the solution is optimized. Transform **5.4.1 Engagement A/I** interfaces with the operator input to insure a proper interface with the **5.4.2 - Order Engagement** transform.

Figure 4-10 is a summary of the DFD development with the details of the individually refined transforms shown. It indicates the concurrency of engagement planning, display, and track management. This was not obvious before the development of the DFD's.

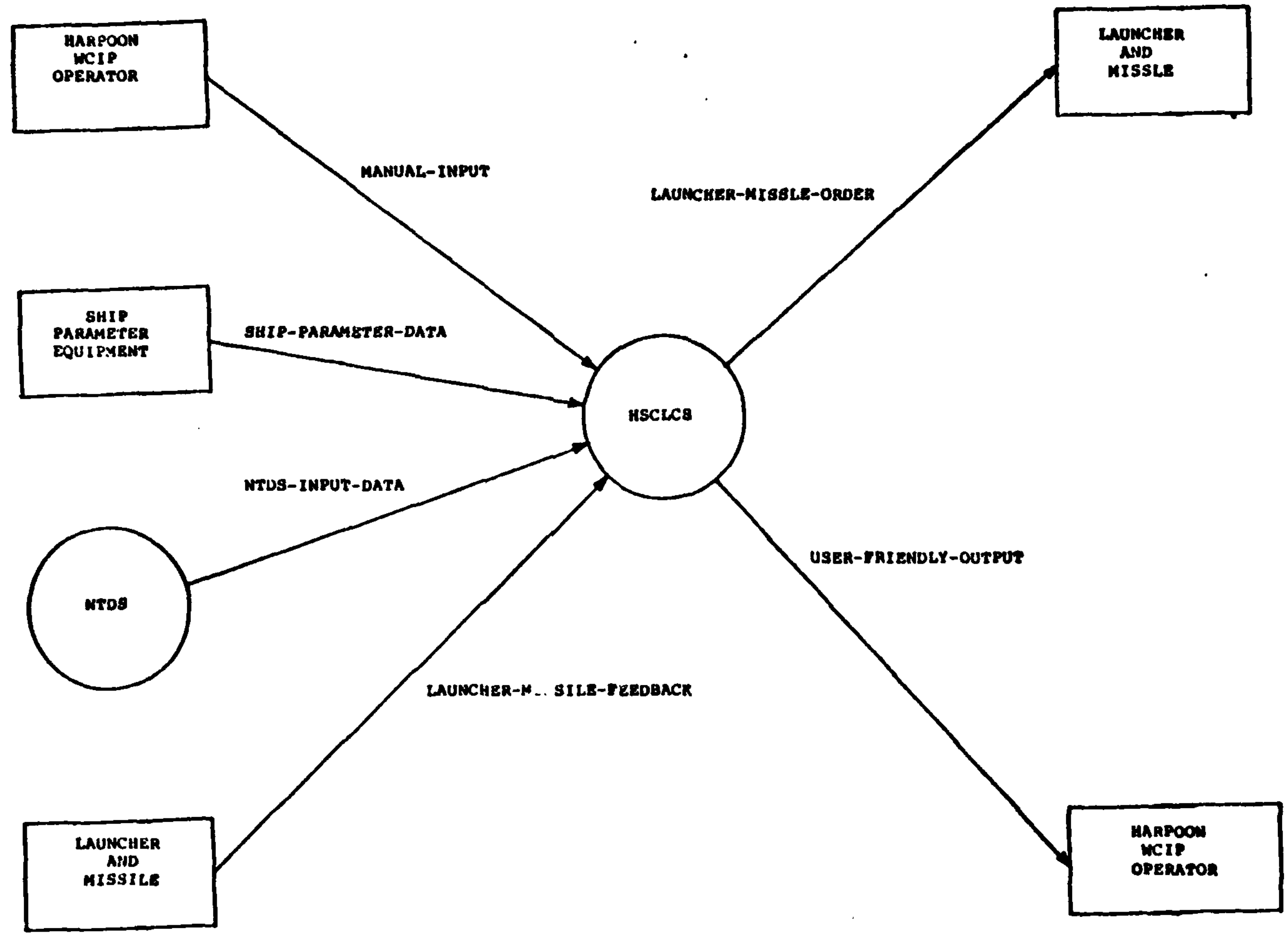

Figure 4-1 Fundamental HSCLCS Systems Model Data Flow Diagram

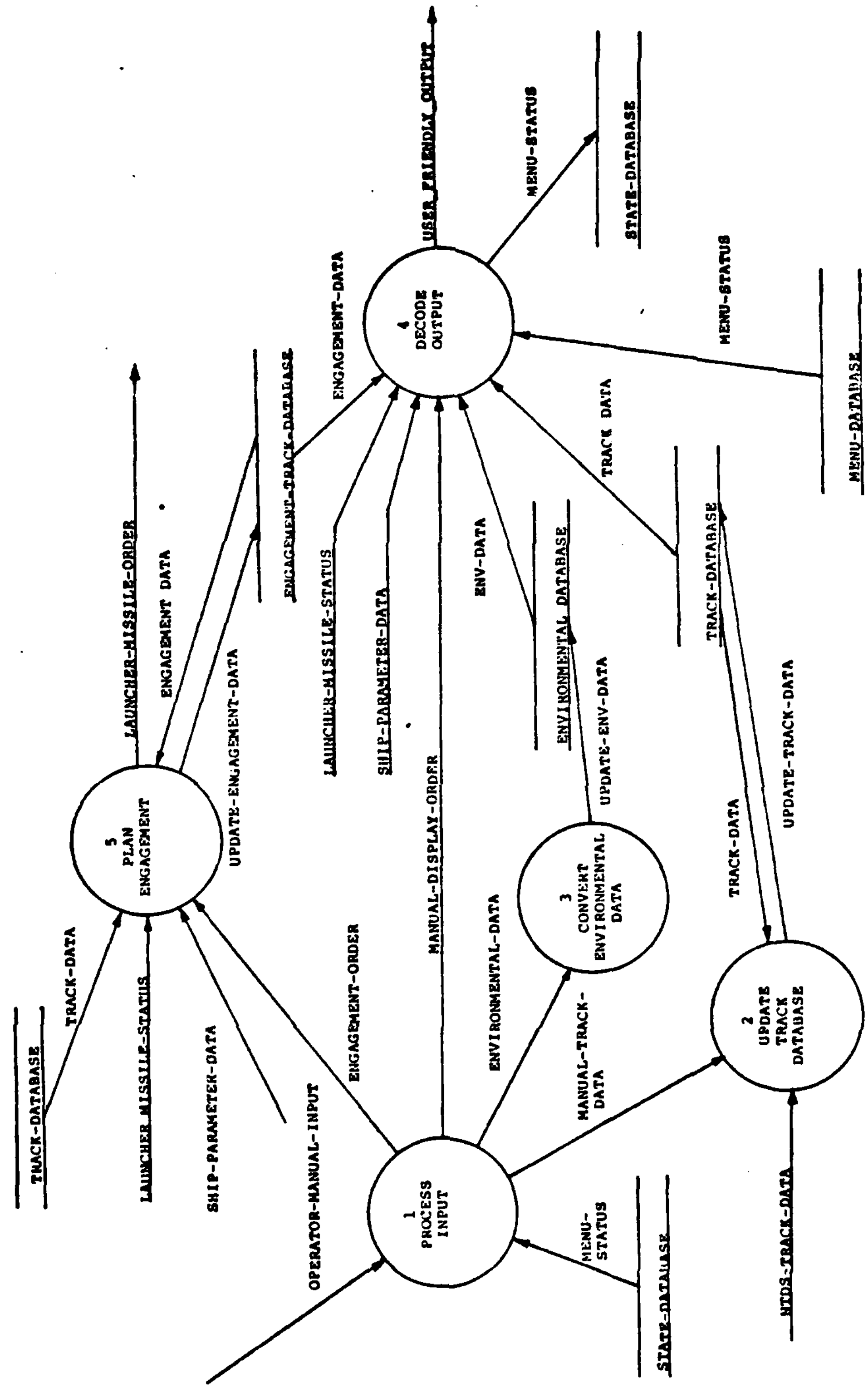

Figure 4-2 HARPOON Weapons System Overview Data Flow Diagram

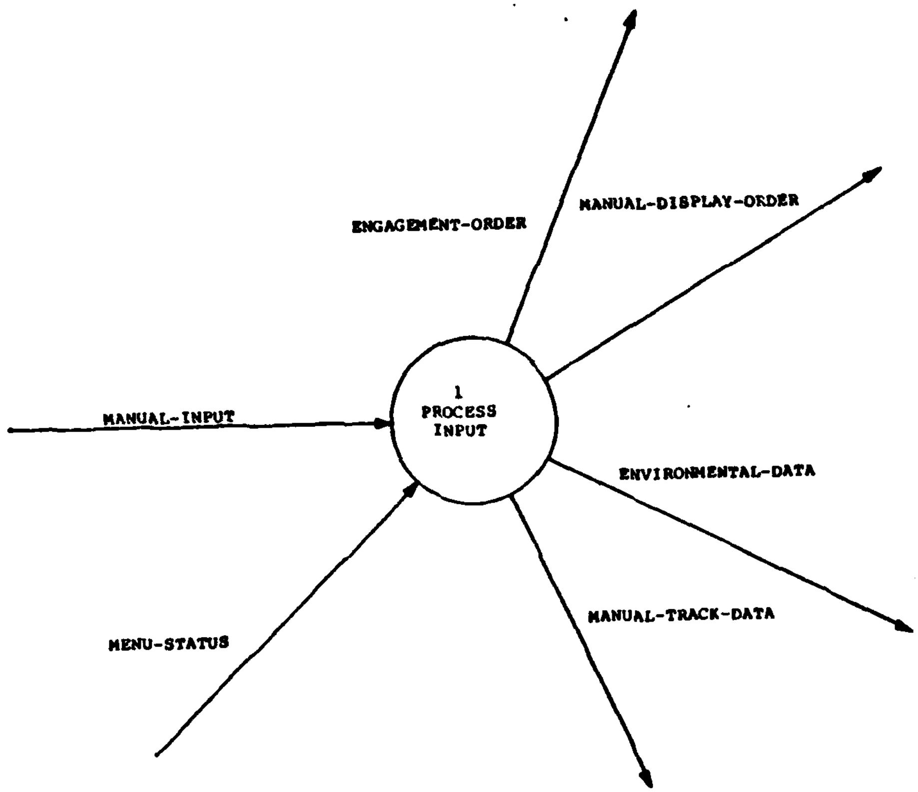

Figure 4-3 Complete Manual Process Data Flow Diagram

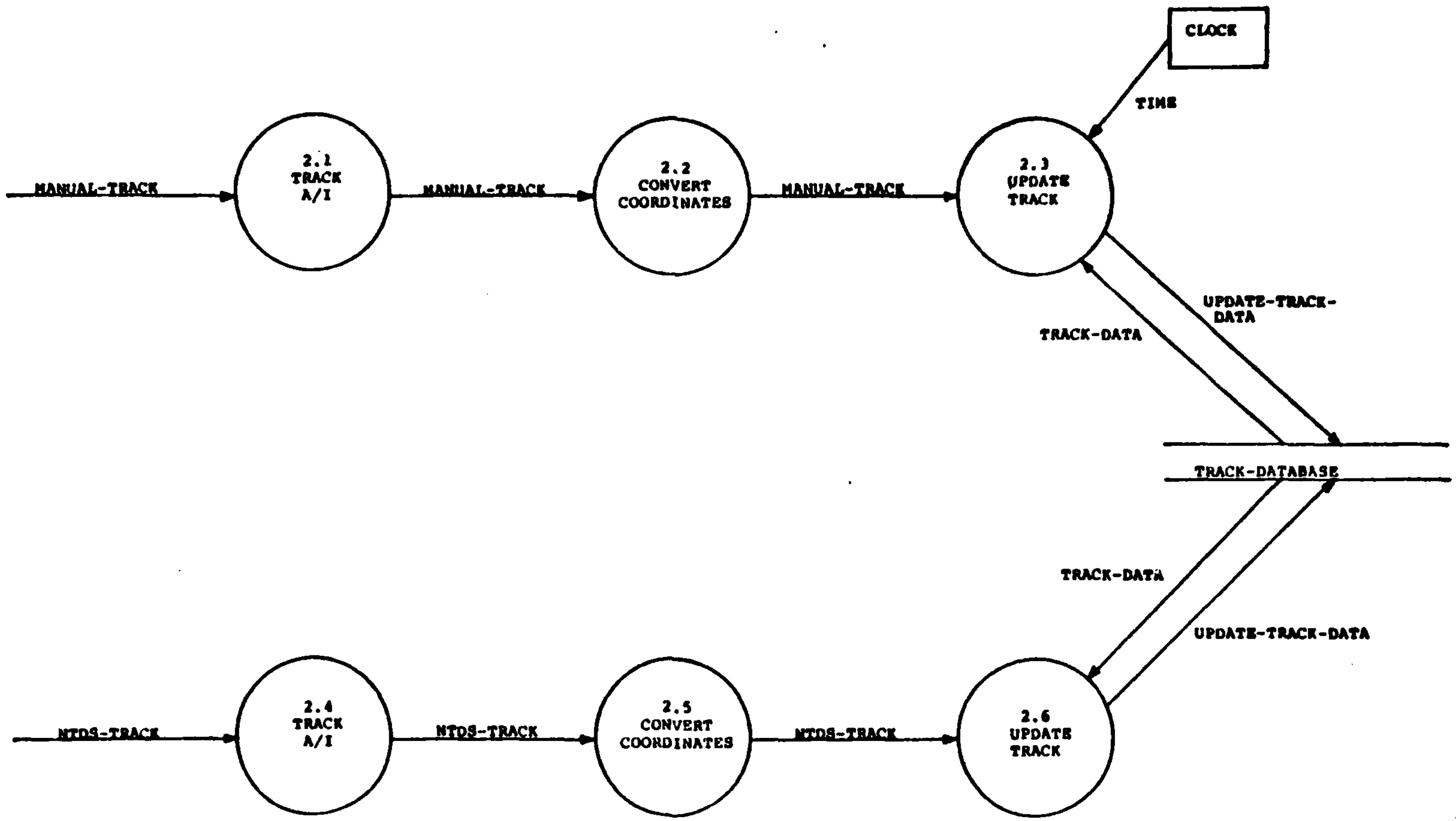

Figure 4-4 Update Track Data Base Data Flow Diagram Refinement Number One

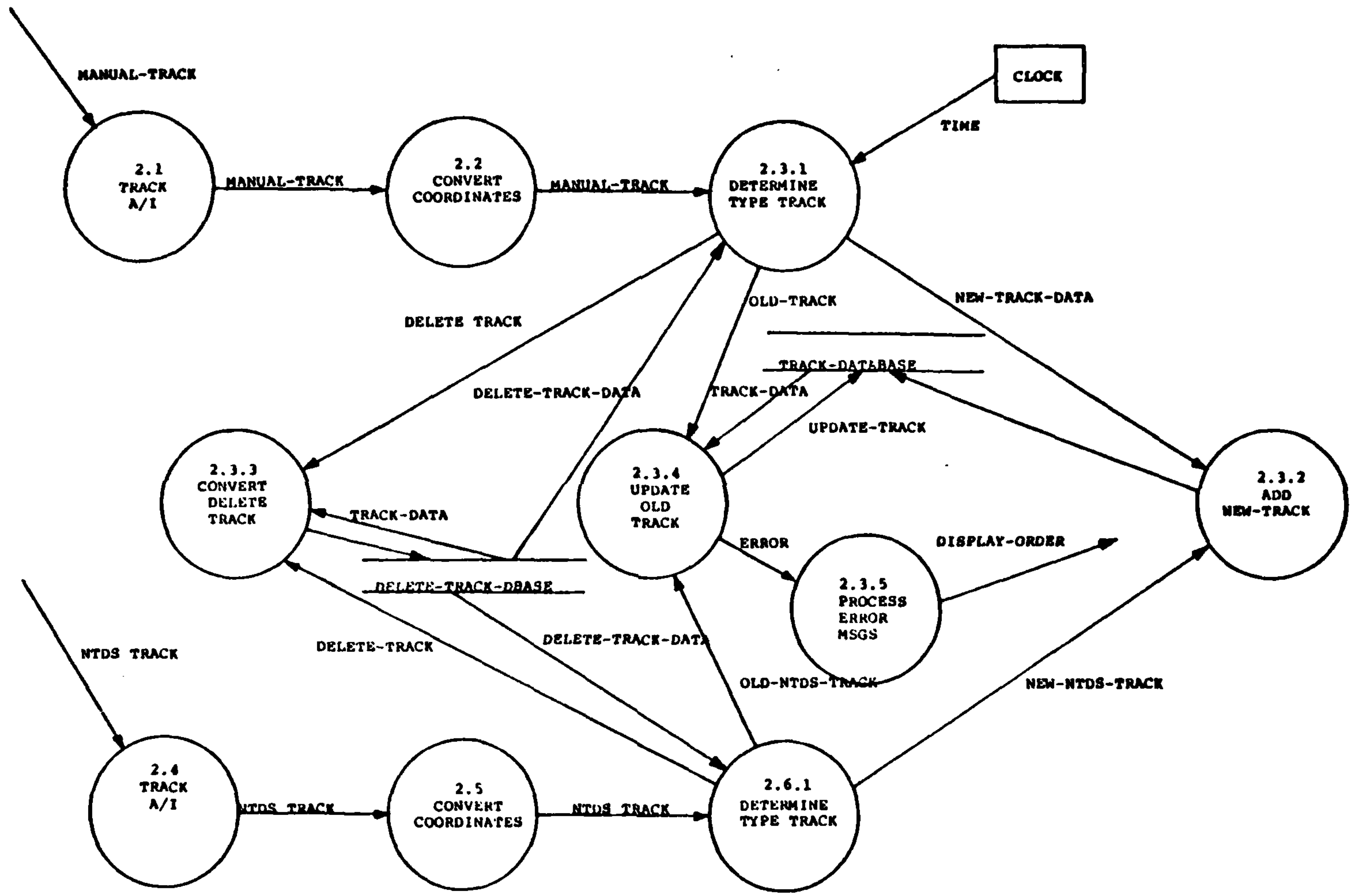

Figure 4-5 Complete Update Track Data Base Refinement Data Flow Diagram

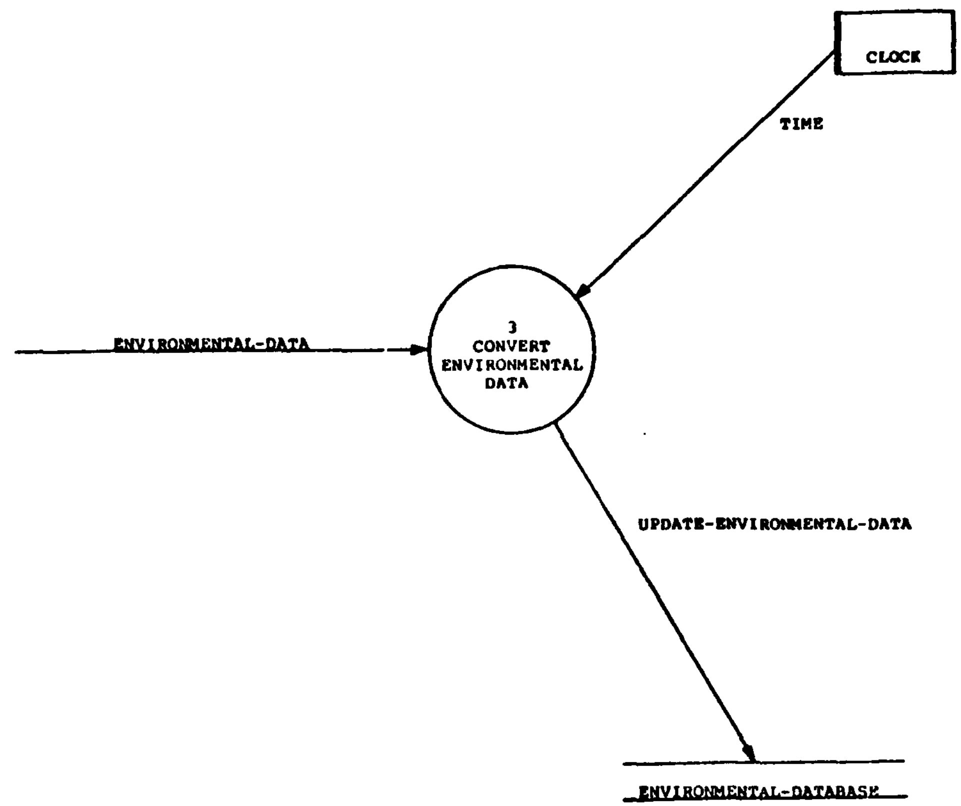

Figure 4-6 Complete Convert Environmental Refinement Data Flow Diagram

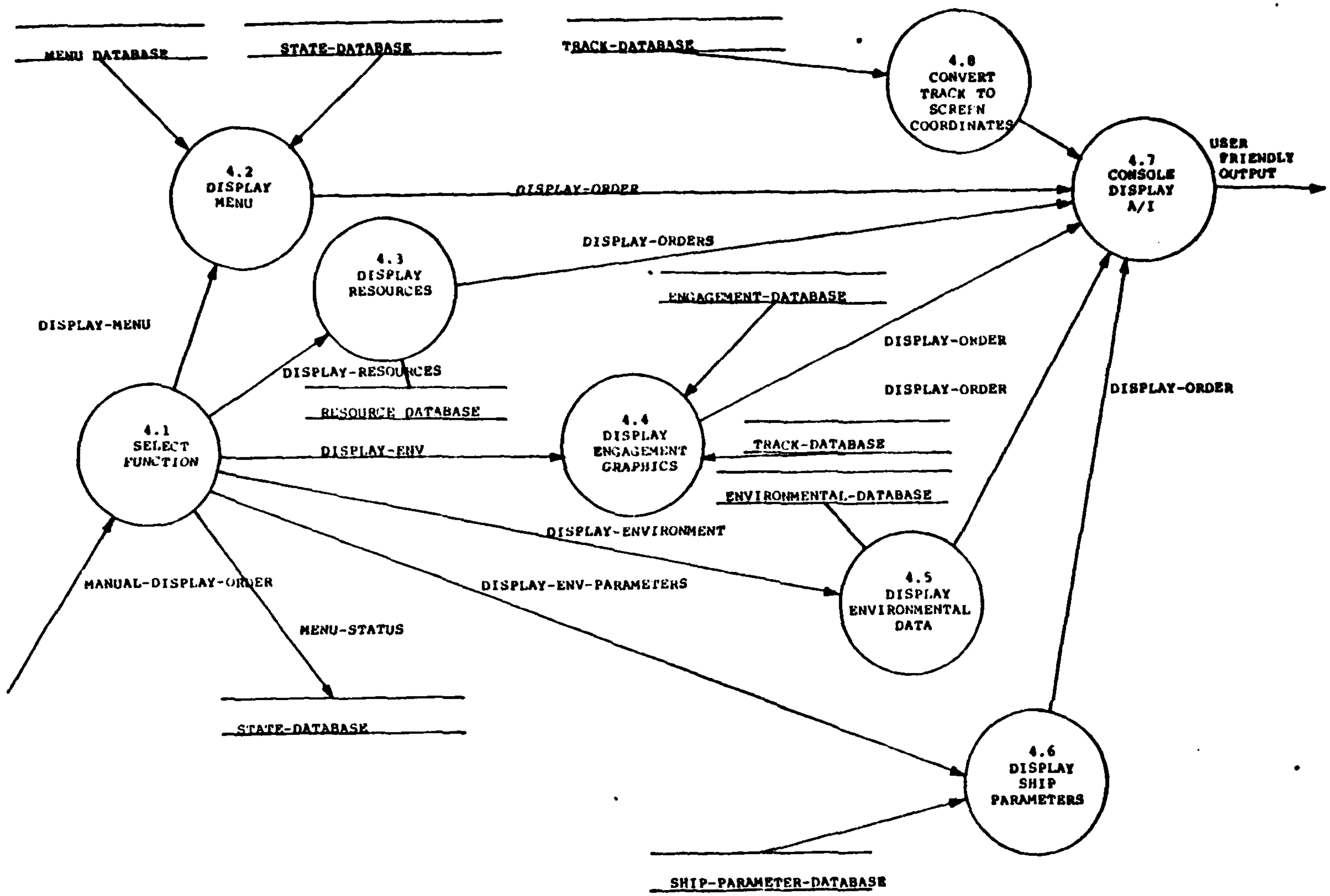

Figure 4-7 Complete Decode Output Refinement Data Flow Diagram

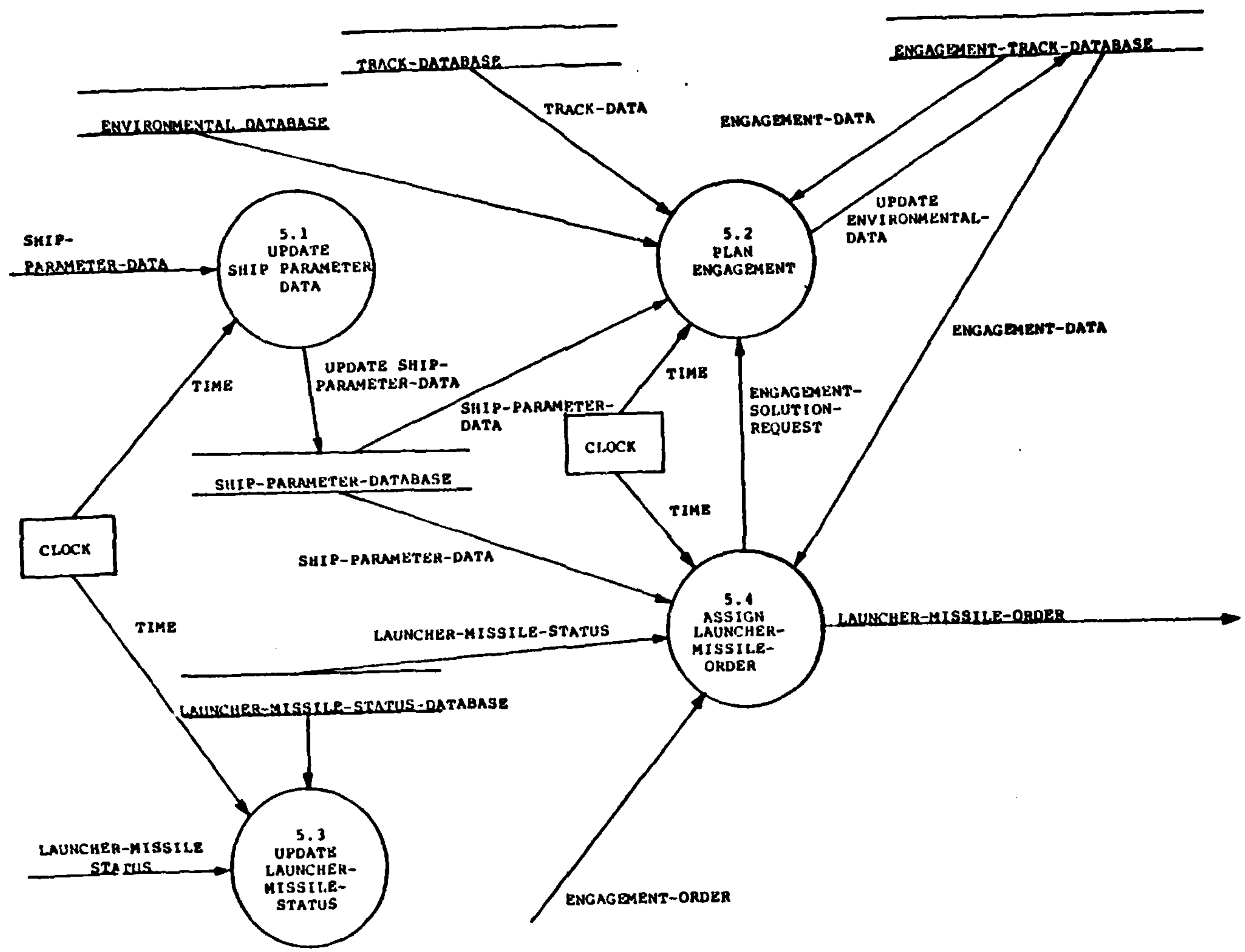

Figure 4-8 Plan Engagement Data Flow Diagram Refinement Number One

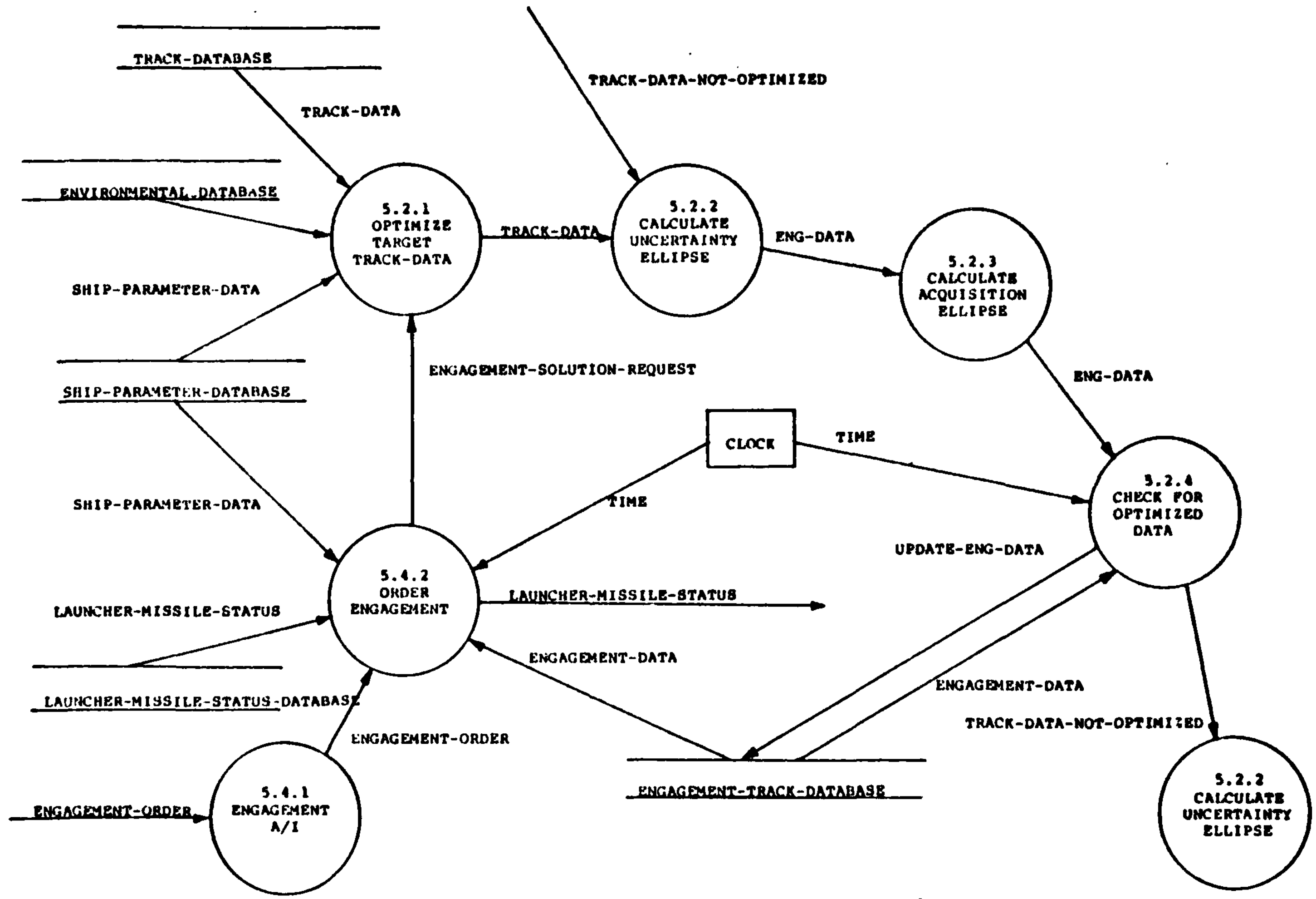

Figure 4-9 Complete Plan Engagement Data Flow Diagram

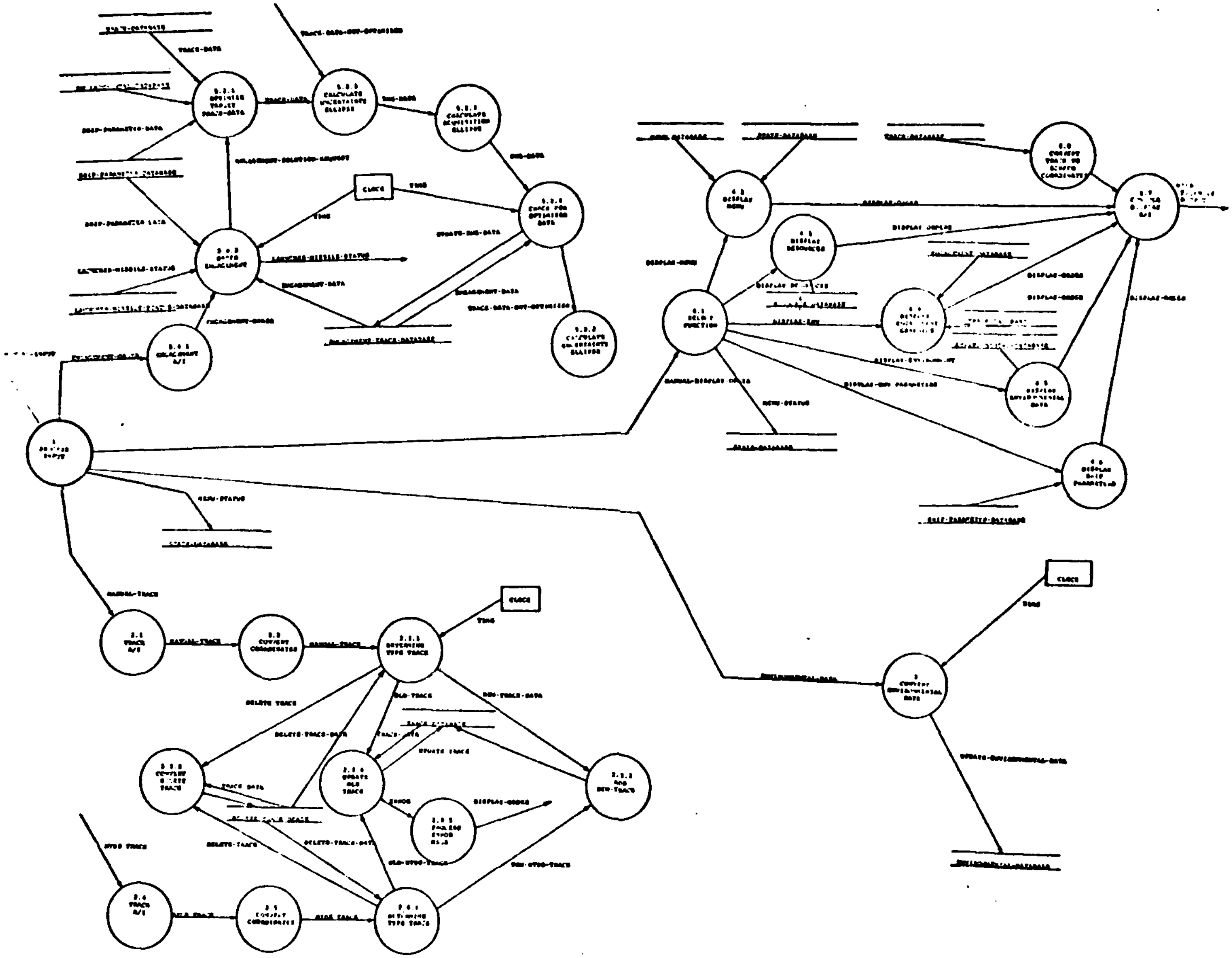

Figure 4-10 Completed HSCLCS System Data Flow Diagram

D. DATA STRUCTURE REPRESENTATION

The data structure of the major databases of the HSCLCS design are detailed in the **Data Structure Definition** format shown in Figure 4-11. These Data Structure Definitions detail the first-cut description of the databases. These descriptions have considerable detail. The eight **Data Structure Definition** representations are contained in Appendix C.

Data Structure Definition

1. Data Structure Name:
2. Data Structure Scope:
3. Data Structure Purpose:
4. Data Structure Users
   a. Write Access:
   b. Read Access:
   c. Read/Write Access:
5. Implementation of Data Structure:
6. Detailed Structure:
7. Operations on Data Structures:
8. Initialization and Range of Data Structure:

   Variable   Initialization   Range
9. Default Value of Data Structure:

Figure 4-11. Sample Data Structure Definition

E. REMAINING PORTIONS OF THE SOFTWARE REQUIREMENTS SPECIFICATION

The following listing of the Software Requirements Specification from Reference 1 were not completed by the authors:

- Information description
  - Data dictionary
  - System interface description
  - Internal interfaces
- Functional description
  - Functions
  - Processing narrative
  - Design constraints
- Validation criteria
  - Performance bounds
  - Classes of tests
  - Expected software response
  - Special considerations
- Bibliography
- Appendix
  - Details of desired algorithms
  - Charts, graphs or other materials
- Preliminary user's manual

V. HSCLCS INFORMATION FLOW TRANSITION TO STRUCTURE

The HSCLCS complete system data flow diagram of Figure 4-10 describes the data flow in a clear, concise manner. The predesign step uses the DFD's to develop the structure diagram of the HSCLCS by performing a mapping, using a combination of the transform and transaction analysis methods which follow.

A. TRANSFORM ANALYSIS

Transform analysis is a set of design steps that allows a DFD with transform flow characteristics to be mapped into a predefined template for software structure. Transform flow is defined as flow that can be characterized by an afferent flow (i.e., incoming data), transformation (i.e., some change or action on the data), and efferent flow (i.e., output flow) with no regard to the number of flow paths [Ref. 1]. The transform analysis method consists of the following seven steps [Ref. 1]:

1. Review of the fundamental system model. Reevaluate the system specifications and the software requirements specification.

2. Review and refine the data flow diagrams. That is, confirm that the overall DFD is detailed enough for a "first cut" design.

3. Determine whether the DFD has transform or transaction characteristics. Look for a distinct transaction center, if there is none assume transform flow exists. A

transaction center is where there are many "action paths" emminating from a single transform. An example of a transaction center is 1 - **Process Input** transform of Figure 4-2.

4. Isolate the transform center by specifying afferent and efferent flow boundaries. Draw a dotted line between the afferent (data flow into) data flow and the transform, and the transform and the efferent (data flow out of) data flow.

5. Perform "first-level factoring". Factoring is a process that distributes control. For transform flow a control module resides at the top to coordinate the three other control modules that control afferent, transform, and efferent flow.

6. Perform "second-level factoring". This is the mapping of the transforms of each afferent, transform, or efferent flow into the subordinate levels of the control structure.

7. Refine the "first-cut" software structure. Use design measures and heuristics to explode or implode modules and thereby produce sensible factoring. The design heuristics are detailed in V. C. REFINING THE HSCLCS SOFTWARE STRUCTURE.

B. TRANSACTION ANALYSIS

Transaction analysis is the analysis which occurs when a transaction triggers one of many possible data flows along two or more paths. A transaction is represented in a DFD as a variety of data flows available at a transction center [Ref. 1]. Steps 1, 2, 3, and 7 are the same as for transform analysis [Ref. 1].

1. Review of the fundamental system model.

2. Review and refine the data flow diagrams.

3. Determine whether the DFD has transform or transaction characteristics.

4. Identify the transaction center. The transaction center is found by determining the bubble origin of a number of radially eminating information paths.

5. Perform "first-level factoring". The transaction flow has a reception branch, a transform control at the apex, and a dispatch flow to which the DFD is mapped.

6. Perform "second-level factoring". Factor and refine the transaction structure and the structure of each action path.

7. Refine the "first-cut" software structure. Use design measures and heuristics for transform analysis.

C. HSCLCS "FIRST-CUT" SOFTWARE STRUCTURE

Figure 5-1 is a duplicate of Figure 4-10 with the addition of dashed lines to indicate where the transform and transaction flow occurs. The completed "first level factoring" of Figure 5-1 is shown in Figure 5-2 while "second level factoring" is shown in Figures 5-3 through 5-6. The control structure is clearly represented by these figures.

D. REFINEMENT OF THE HSCLCS SOFTWARE STRUCTURE

The refinements to the software structure require the use of design heuristics. These heuristics are developed from the best of current thought on the software design process. These heuristics from Reference 1 are detailed below. Figure 5-7 is an example of a refinement to the HSCLCS structure using the first of the following heuristics on Figure 5-3.

- "Evaluate the preliminary software structure to reduce coupling and improve cohesion." Look critically at the developed structure to see if modules should be "exploded" so parts of them can be shared by other modules requiring the same function or "imploded" if the process is only done in a particular module.

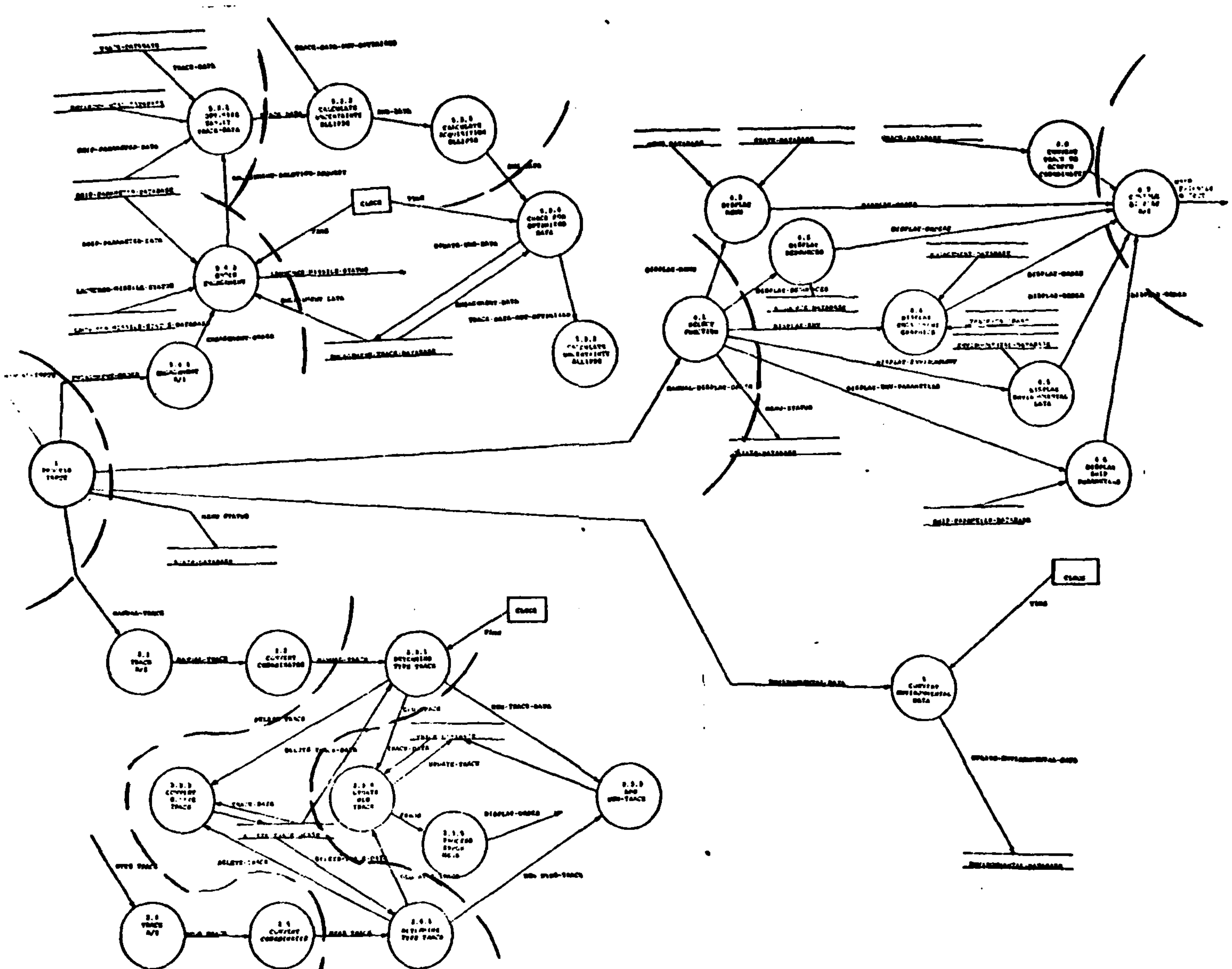

Figure 5-1 Complete HSCLCS System Data Flow Diagram

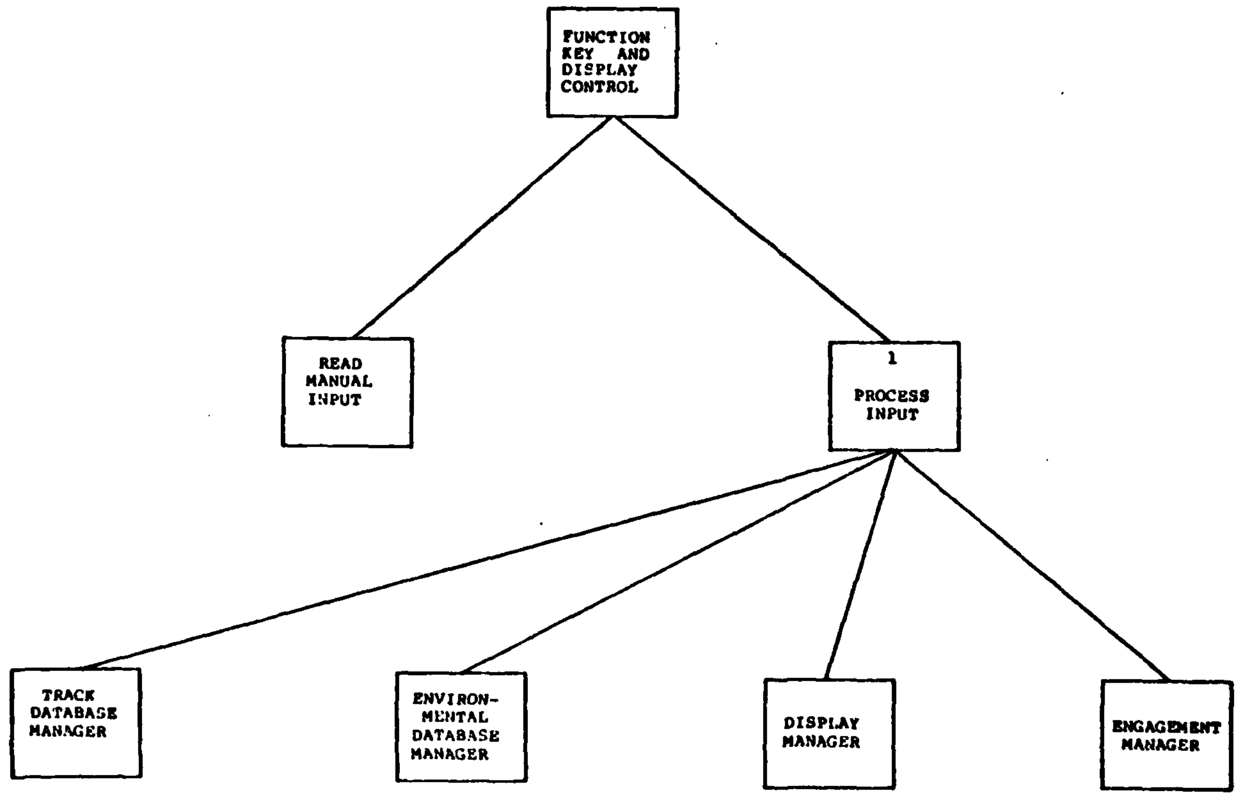

Figure 5-2 Mapping of the Transition Structure of Figure 5-1

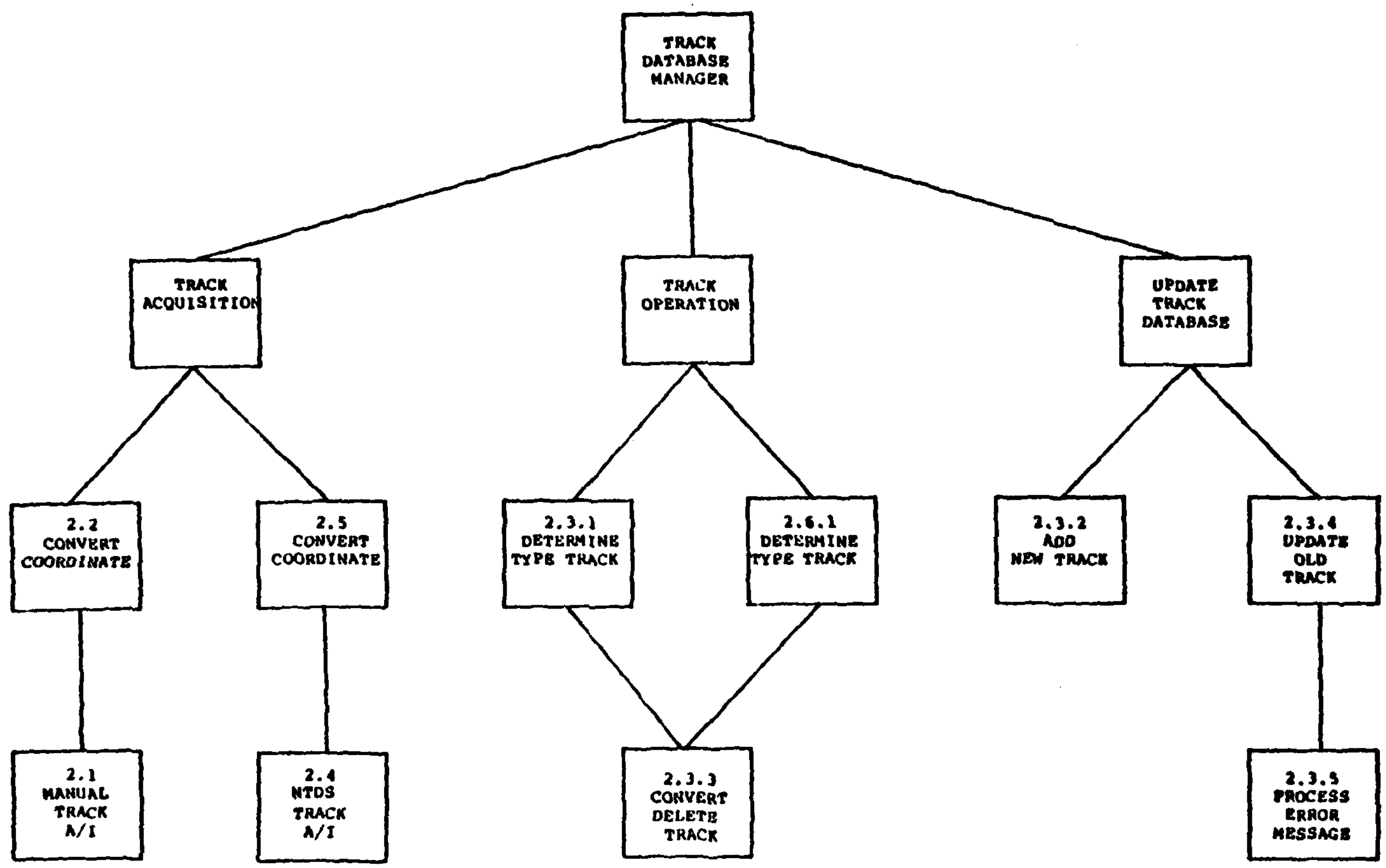

Figure 5-3 Action Path Structure of Track Data Base Manager

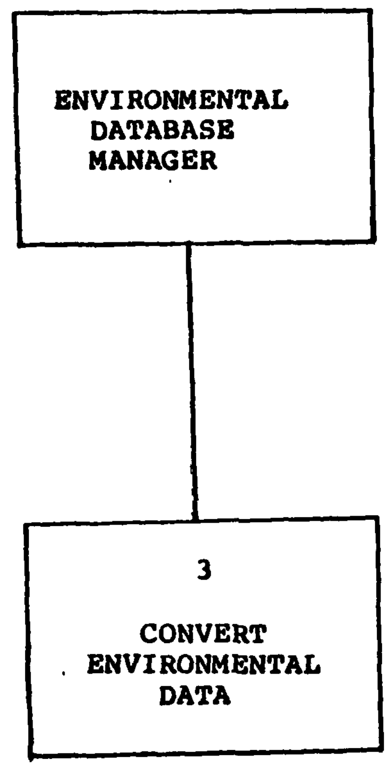

Figure 5-4 Action Path Structure of Environmental Data Base Manager

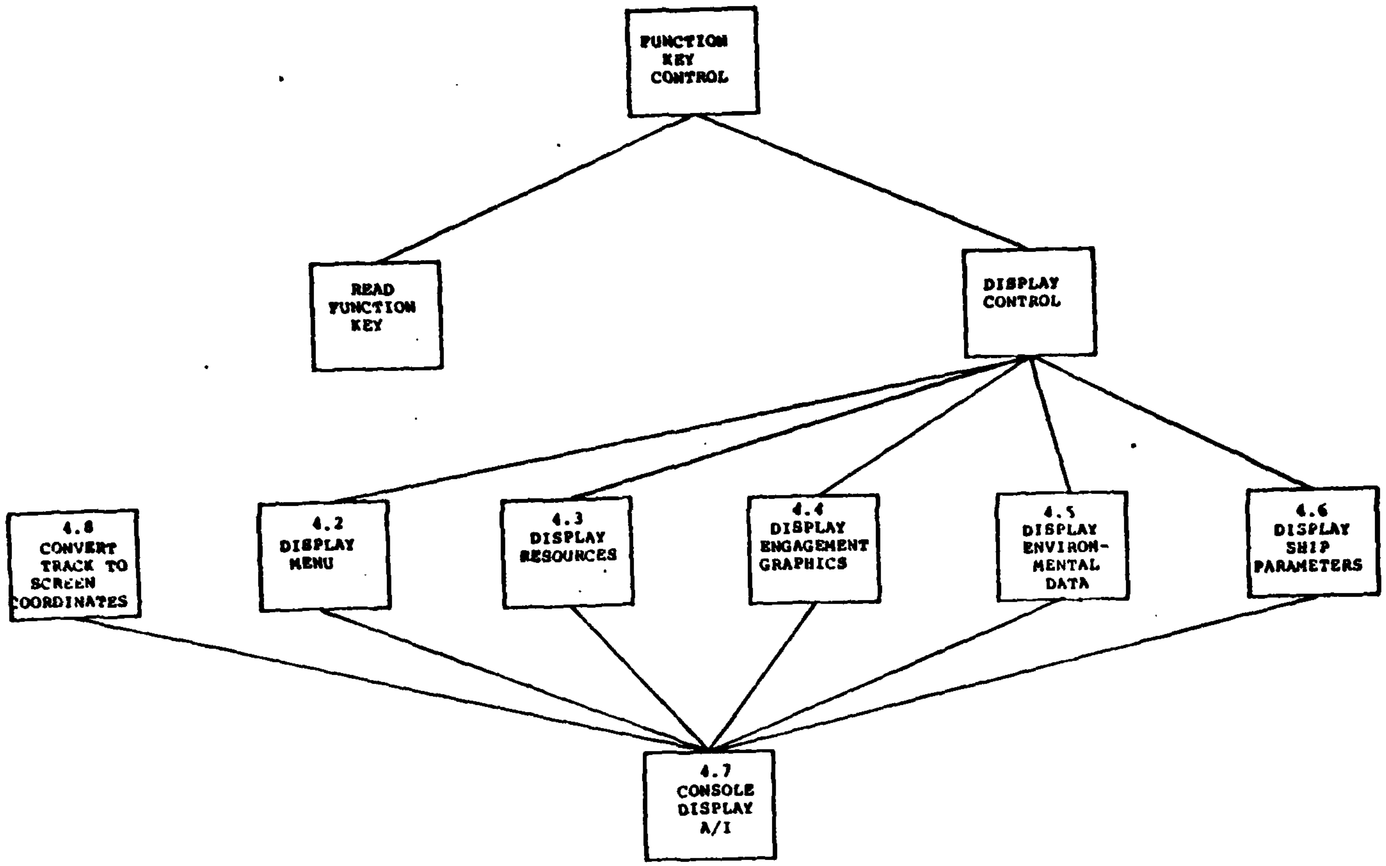

Figure 5-5 Action Path Structure of Display Manager

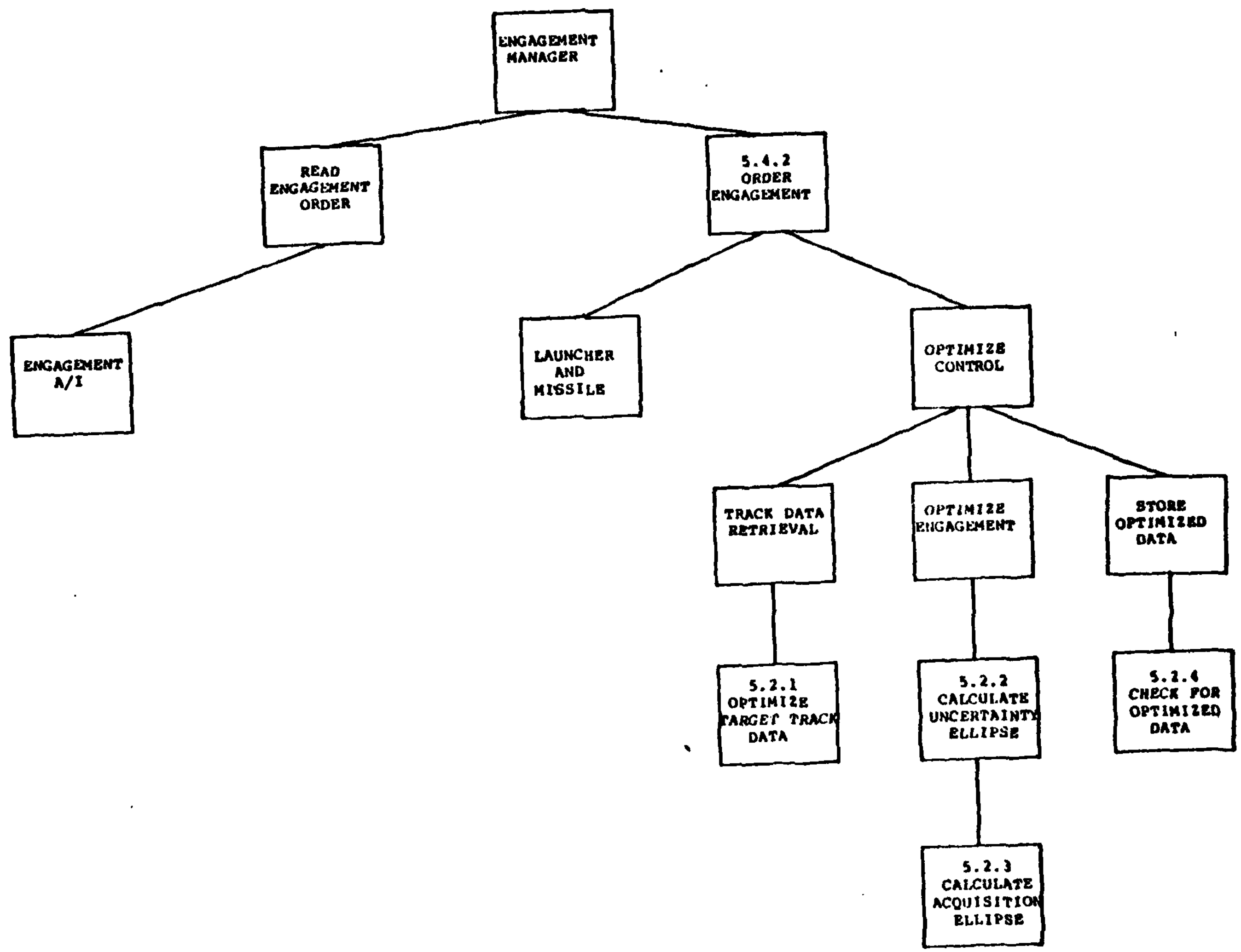

Figure 5-6 Action Path Structure of Engagement Manager

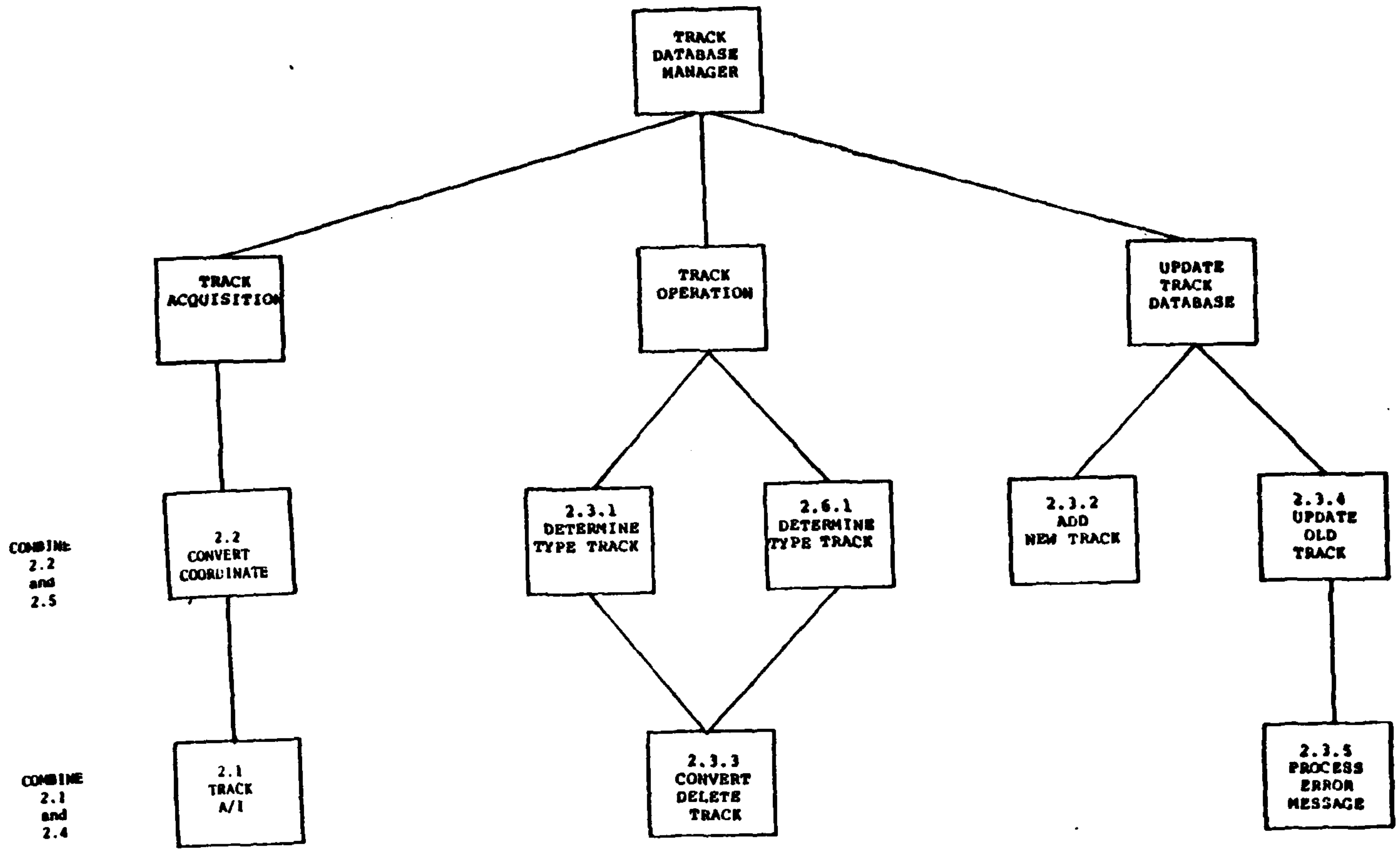

Figure 5-7 Action Path Structure of Track Data Base Manager, With Heuristic

- "Attempt to minimize structures with high fan-out; strive for fan-in as depth of the structure increases." The ideas of "fan-out" and "fan-in" concern effective factoring. By having a narrower, deeper (fan-in vice fan-out) structure there are a number control layers, reminiscient of transform vice transaction analysis. This is a "more reasonable distribution control.

- "Keep scope of effect of a module within the scope of control of that module." For a module, all of the modules that are within its "scope of control" should be the only ones that are also in its "scope of effect" (i.e. only modules subordinate to the module or lower in the structure).

- "Evaluate module interfaces to reduce complexity and redundancy and improve consistency." Insure passing of the proper parameters is made. Parameter passing should be clear and logical.

- "Define modules whose function is predictable, but avoid modules that are overly restrictive." "A module is 'predictable' when it can be treated as a 'black box'." This is the principle of "information hiding" that is with given inputs the outputs remain the same regardless of the operations that occur in the "black-box".

- "Strive for single-entry, single-exit modules, avoiding "pathological connections." This is the prohibition of "goto's" to prevent "pathological connection" which refers to branches or references in to or out of the middle of a module.

- "Package software based on design constraints and portability requirements." Packaging alludes to the techniques used to assemble software for a specific processing environment or to ship software to a remote location.

- "Select the size of each module so that independence is maintained." i.e. approximately 50 lines of code.

E. DESIGN POSTPROCESSING

After the structure is developed by using the transform/transaction technique and then factored further by

the use of design heuristics, design postprocessing is required, which consists of [Ref. 1]:

- A processing narrative which must be developed for each module.
- An interface description for each module.
- Definition of local and global data structures.
- Notation of design restrictions and limitations.
- A preliminary design review.
- "Optimization" of the design as desired.

"First-cuts" at the post processing narrative for each module and interface description are detailed in Appendix D using the format of Figure 5-8.

Module Description

1. Module Name:
2. Module Purpose:
3. Module Interface Definition
   a. Inputs:
   b. Outputs:
   c. Miscellaneous
4. Module Processing Narrative Description:
5. Superordinate Modules:
6. Subordinate Modules:
7. Error Detection/Handling:
8. Design Decisions:
   a.

Figure 5-8. Sample of Module Description

F. REMAINING REQUIREMENTS OF THE SOFTWARE ENGINEERING PROCESS

There are several design steps that remain and follow from the preliminary design which are not pursued by the authors. These design requirements in general are (refer to Figure 1-1):

- Detailed design.
- Code and unit testing.
- Testing.

VI. CONCLUSIONS AND RECOMMENDATIONS

Chapter II. System Specification detailed the need for a redesign of the HSCLCS to take full advantage of the HARPOON missile capabilities of Block 1C and newer versions. The HSCLCS desirable improvements were discussed in detail in the same chapter and are the basis for the design effort of this report. Chapter IV and V detail the software engineering design effort used in modeling the system specifications, and should leave the reader with the impression that the system specifications can be designed into a working, upgraded HSCLCS. The following are the primary design characteristics resulting from the HSCLCS redesign:

- Implement a user friendly plasma display that allows operator inputs to drive the "state" of the system so that programmable function buttons may be used for a variety of software controlled tasks.

- Develop a track data base for both manual operator input and external combat systems inputs (i.e. SONAR, fire control radar, NTDS, etc.)

- Perform automatic engagement analysis on hostile tracks (as well as any operator designated track). Accomplish these calculations when the CPU is not doing a higher priority task. The implication here is that a carefully thought out algorithm is the key to an effective , if not optimal automatic engagement algorithm.

A. RECOMMENDED FOLLOW-ON WORK

The authors recommend research be conducted at Naval Postgraduate School in the following areas in support of HSCLCS improvements:

- Continue the redesign effort commencing from the initial design effort of this report.
- Examine the interface requirements for all existing HARPOON launch systems.
- Use Ada as the developmental language and code the completed design.
- Research the current verification and validation practices for this and future software engineering projects.
- Explore the development of graphic techniques to enhance the human engineering aspect of the HSCLCS design.
- Discuss the design aspects of HARPOON that are directly transferable to the Tomahawk cruise missile and other cruise missile follow-ons.

APPENDIX A

GLOSSARY

Abstraction - A psychological notion that affords a view of a problem at some level of generalization without regard to irrelevant low level details. Use of abstraction allows the use of concepts and terms that are familiar in the problem environment without having to transform them to an unfamiliar environment [Ref. 1].

Abstract Interface - Allows inputs into or outputs from a module to match changes in inputs or outputs to only affect the abstract interface code, and not the code on the output side of the module. Trys to solve the embedded computer problem and keep the cost down.

Bubble Diagram - see Data Flow Diagram (DFD).

Cohesion - A measure of the relative functional strength of a module. Best to describe as a spectrum from 'single-minded' to 'scatter-brained' (i.e. variety of functions performed) [Ref. 1].

Correctness - A program is correct if it performs properly the functions it was intended (specified) to do and has no unwanted side effects [Ref. 1].

Coupling - A measure of the relative independence among modules. Varies from no direct coupling to content coupling,

i.e. one module uses data or control information maintained within the boundary of another module [Ref. 1].

Data Flow Diagram (DFD) (sometimes called a bubble diagram)- A graphical tool used to depict data (information) flow. The DFD uses the following graphical symbols: labeled arrows to represent information flow, labeled circles called "bubbles" that represent processes (transformations), labeled boxes that represent information sources and sinks, and two labeled parallel lines that represent stored information [Ref. 1].

Data Base- A file of interrelated data that are stored together to serve one or more applications and that are independent of programs using the data [Ref. 2].

Data Dictionary - A repository of information about data and process associated with a particular systems development effort. Includes a glossary of terms, data characteristics, process description, and cross references [Ref. 2].

Data Structure - Dictates the organization, methods of access, degree of associativity, and processing alternatives for information [Ref. 1].

Debug- To detect and to correct errors in a procedure, system, process, or module, or in a piece of equipment [Ref. 2].

Deliverable - The output required at the end of some portion of the software engineering cycle. The reason for "that" particular software design step [Ref. 1].

Domain of Change - Area of a system that is 'subject to change' or 'subject to future change'. Related to DFD [Ref. 1].

Embedded Program - A computer program that is part of some larger entity and essential to the operations of that system. For example, the timer on a washing machine or the guidance system in a missile may have computer programs. These programs are considered to be *embedded*.

Execute - To carry out an instruction or to perform a routine or set of routines [Ref. 2].

Fan-out - A measure of the number of modules that are directly controlled by another module [Ref. 1].

Fan-in - Indicates how many modules directly control a given module [Ref. 1].

Flowchart - A graphical tool to show sequence and control of program or module logic [Ref. 2].

Function - Name given to one or more statements that perform a specific task. Results in a value being assigned to its name upon execution of that function [Ref. 1].

Functional Allocation - Optional allocations of where a particular design function should be performed (i.e.,

hardware or software implementation desired). One selection is selected from the alternatives [Ref. 1].

Information Hiding - Specification and design of modules so that information (procedure and data) contained within a module are inaccessible to other modules that have no need to know the information [Ref. 1].

Interface - Communications between modules governed by a set of assumptions one module makes about another [Ref. 1].

Maintenance - The phase in a system's life cycle following development, acceptance, and installation [Ref. 2].

Module - A separately addressable element of a program [Ref. 1].

Modular Design - A logical partitioning of software into elements that perform specific functions or subfunctions [1].

Packaging - Alludes to the techniques used to assemble software for a specific processing environment or to ship software to a remote location [1].

Pathological Connections - Refers to branches or references into or out of the middle of a module [1].

Predictable Module - One that can be treated as a black box; that is, the same external data will be produced regardless of initial processing details [Ref. 1].

Procedure - A subprogram within a program [Ref. 1].

<u>Procedure Name</u> - Defines a block of statements that will be executed as a program every time the name of the procedure is invoked [Ref. 1].

<u>Recursion</u> - Procedure name may be invoked within the procedure definition; that is, a procedure may call itself. This may be an expensive procedure. Recursion is a characteristic available in PL/I, Pascal, and Ada. It often makes programming easier and programs easier to understand [Ref. 1].

<u>Requirements Analysis</u> - Third step in the software engineering procedure, last of the planning phase steps. Provides a foundation for the development of the software by uncovering the flow and structure of information. Describes the software by identifying interface details, providing an in-depth description of functions; determining design constraints and defining software validation requirements. Establishes and maintains communication with the user and the requester so that the above two objectives may be satisfied [Ref. 1].

<u>Robustness</u> - A program is robust if it will continue to do something reasonable in the presence of software environmental changes (such as hardware failure) and demands (such as data) that were not foreseen [Ref. 1].

Scope-of-control - Contains all the modules that are subordinate and ultimately subordinate to the module [Ref. 1].

Scope-of-effect - The range of modules that are effected by a module decision [Ref. 1].

Single-entry - Only one way to enter a module [Ref. 1].

Single-exit - Only one way to exit a module [Ref. 1].

Software Engineering - Software implementation of a problem solution approached by using a set of techniques that are application independent. These techniques are (1) a well-defined methodology that addresses a software lifecycle of planning, development, and maintenance, (2) an established set of software components that documents each step in the life cycle and shows traceability from step to step, and (3) a set of predictable milestones that can be reviewed at regular intervals throughout the software lifecycle [Ref. 1].

Software Requirements Specification - The deliverable of the software requirements analysis phase of the software engineering process. Contains introduction, information description, functional description, validation criteria, bibliography, and appendix [Ref. 1].

Software Plan - Second step in the software engineering process. Provides a framework that enables the manager to make reasonable estimates of resources, cost, and schedule.

Stepwise Refinement - The architecture of the program is developed by successively refining levels of procedural detail. Early software top-down design procedure proposed by Niklaus Wirth [Ref. 1].

Subordinate Module - A module controlled by another module [Ref. 1].

Superordinate Module - A module that controls another module [Ref. 1].

System - A collection of elements related in a way that allows accomplishment of some tangible objective [Ref. 1].

System Analysis - Comprised of a number of tasks that define what must be accomplished, whether accomplishment is feasible, and what the cost-benefit of accomplishment will be [Ref. 1].

System Definition - First step in the software planning phase and an element of the computer system engineering process described in chapter 1. Attention is focused on the system as a whole. Functions are allocated to hardware, software, and other system elements based on a preliminary understanding of requirements. Comprised of three tasks: systems analysis, functional allocation, and system specification [Ref. 1].

System Specification - First deliverable in the computer system engineering process. Contains introduction, functional description, allocation, constraints, cost,

schedule of system development known at the time of the completion of the system specification [Ref. 1].

<u>Transaction</u> <u>Flow</u> - A variety of data flows available at a transaction center [Ref. 1].

<u>Transform</u> <u>Flow</u> - Flow that can be characterized by an afferent flow (i.e., incoming data), transformation (i.e., some change or action on the data, and efferent flow (i.e., output flow) with no regard to the number of flow paths [Ref. 1].

APPENDIX B

ACRONYMS AND ABBREVIATIONS

BIT - Built-In Test

BITE - Built-In Test Equipment

BOL - Bearing Only Launch

BRG - Bearing

BSTR - Booster

C&C - Command and Control

CP - Casualty Panel

CIC - Combat Information Center

DCU - Data Conversion Unit

DPC - Data Processor Computer

EAS - Engagement Analysis System

EPS - Engagement Planning System

FCS - Fire Control System

HSCLCS - HARPOON Ship Command-Launch Control Set

HWS - HARPOON Weapons System

LSU - Launcher Switching Unit

NTDS - Naval Tactical Data Systems

RNSH - Royal Navy Sublaunched HARPOON

RBL - Range Bearing Launch

RMS - Resource Management System

TDBMS - Track Data Base HARPOON Management System

TDS - Tactical Data System

UBFCS - Underwater Battery Fire Control System

WCIP - Weapon Control Indicator Panel

WCS - Weapon Control System

APPENDIX C

HSCLCS DATA STRUCTURE DEFINITIONS

This appendix contains a first-cut HSCLCS data structure definition of the details on the eight data bases derived during the development of the DFD's. The eight data bases are:

1. Environmental data base
2. Menu data base
3. Engagement data base
4. Track data base
5. Delete track data base
6. Ship parameter data base
7. Launcher missile status data base
8. State data base

Data Structure Definition

1. Data Structure Name: /env state/ Environmental data base.

2. Data Structure Scope: Global.

3. Data Structure Purpose: Contains weather variables of: /sea state/;/rain state/;/wind state/.

4. Data Structure Users

   a. Write Access: 3 Convert Environmental Data

   b. Read Access: 5.2.1 Optimize Target Track Data
   4.5 Display Environmental Data

   c. Read/Write Access: None.

5. Implementation of Data Structure: Pointer based record.

6. Detailed Structure: /env state/ = record;

   /find/ : ptr;

   /sea state/ : integer;

   /rain state/ : boolean;

   /wind state/ : integer.

7. Operations on Data Structures: Updates only by HSCLCS operator. Used for engagement calculations as a comparison value.

8. Initialization and Range of Data Structure:

Variable	Initialization	Range
/sea state/	0	0 to 10 (Beaufort)
/rainstate/	NO	YES/NO
/windstate/	0	0 to 100 (knots)

9. Default Value of Data Structure: Same as 8. initialization.

Data Structure Definition

1. Data Structure Name: /menu/ Menu data base.

2. Data Structure Scope: Global.

3. Data Structure Purpose: Change the menus associated with the programmable function buttons

4. Data Structure Users

   a. Write Access: No write access, pre-programmed

   b. Read Access: 4.2 Display Menu

   c. Read/Write Access: None

5. Implementation of Data Structure: Pointer based record.

6. Detailed Structure: Undetermined

7. Operations on Data Structures: Read to the appropriate section of the plasma display for user function button description.

8. Initialization and Range of Data Structure: Undetermined

Variable	Initialization	Range

9. Default Value of Data Structure: Undetermined

Data Structure Definition

1. Data Structure Name: /engagement/ Engagement data base.

2. Data Structure Scope: Global.

3. Data Structure Purpose: Provide information on the 'optimized' engagement of all hostile tracks.

4. Data Structure Users

   a. Write Access: 5.2.4 Check for Optimized Data

   b. Read Access: 5.2.4 Check for Optimized Data
   5.4.2 Order Engagement

   c. Read/Write Access: 5.2.4 Check for Optimized Data

5. Implementation of Data Structure: Pointer based record.

6. Detailed Structure: Undetermined

7. Operations on Data Structures: /Add/update/delete/use

8. Initialization and Range of Data Structure: Undetermined

Variable	Initialization	Range

9. Default Value of Data Structure: Straight line shot data.

Data Structure Definition

1. Data Structure Name: /track/ Track data base.

2. Data Structure Scope: Global.

3. Data Structure Purpose: Provide means for HSCLCS operator to maintain all the tracks he desires from manual, NTDS (if available) and other tracking sources.

4. Data Structure Users

   a. Write Access: 2.3.4 Update old track and 2.10 Add new track

   b. Read Access: 2.3.4 Update Old Track
      4.8 Convert Track to Screen Coordinates
      5.2.1 Optimize Target Track Data

   c. Read/Write Access: 2.3.4 Update Old Track

5. Implementation of Data Structure: Pointer based record.

6. Detailed Structure: Must contain track number, position information, time of update, type of track, and is pointer based record.

7. Operations on Data Structures: Add/update/delete/use.

8. Initialization and Range of Data Structure: Undetermined.

Variable	Initialization	Range

9. Default Value of Data Structure: Undetermined.

Data Structure Definition

1. Data Structure Name: /delete track/ Delete track data base.
2. Data Structure Scope: Local to track system only.
3. Data Structure Purpose: Store track numbers of tracks not desired to be stored in the track data base.
4. Data Structure Users

   a. Write Access: see c.

   b. Read Access: 2.6.1 Determine Type Track
   2.3.1 Determine Type Track
   2.3.3 Convert Delete Track

   c. Read/Write Access: 2.3.3 Convert Delete Track

5. Implementation of Data Structure: Pointer based record.
6. Detailed Structure:

```
/delete track/ = record;
    /next/ : ptr;
    /track_nr/ : integer.
```

7. Operations on Data Structures: Add/use/update/delete.
8. Initialization and Range of Data Structure:

Variable	Initialization	Range
/track_nr/	empty	0 - 1000

9. Default Value of Data Structure: Not applicable

Data Structure Definition

1. Data Structure Name: /ship_par/ Ship parameter data base.

2. Data Structure Scope: Global.

3. Data Structure Purpose: Retain ship parameter info.

4. Data Structure Users

   a. Write Access: None.

   b. Read Access: 5.2.1 Optimize Target Engagement
   5.4.2 Order Engagement
   4.6 Display Ship Parameters

   c. Read/Write Access: None

5. Implementation of Data Structure: Pointer based record.

6. Detailed Structure:
```
/ship_par/ = record;
/heading/ : float;
/roll/ : float;
/pitch/ : float;
/yaw/ : float;
/time/ : integer;
/point/ : ptr.
```

7. Operations on Data Structures: Add/update/use.

8. Initialization and Range of Data Structure: None required

   Variable Initialization Range

9. Default Value of Data Structure: None.

Data Structure Definition

1. Data Structure Name: /missile stat/ Launcher missile status data base.

2. Data Structure Scope: Global.

3. Data Structure Purpose: Provide user information on status of missile and launcher.

4. Data Structure Users

   a. Write Access: None.

   b. Read Access: 5.4.2 Order engagement.
      4.9 Display launcher missile status.

   c. Read/Write Access: None.

5. Implementation of Data Structure: Single record.

6. Detailed Structure: Undetermined.

7. Operations on Data Structures: Add/use/update/

8. Initialization and Range of Data Structure:Undetermined.

Variable	Initialization	Range

9. Default Value of Data Structure: None.

Data Structure Definition

1. Data Structure Name:/state/ State data base

2. Data Structure Scope: Global.

3. Data Structure Purpose: For use in determining the state of the program so manual inputs can be decoded properly.

4. Data Structure Users

   a. Write Access: 4.1 Select function

   b. Read Access: 1 Process input

   c. Read/Write Access: None.

5. Implementation of Data Structure: Single record.

6. Detailed Structure: Undetermined.

7. Operations on Data Structures: Add/update/use/

8. Initialization and Range of Data Structure: Undetermined.

Variable	Initialization	Range

9. Default Value of Data Structure: Undetermined.

APPENDIX D

HSCLCS MODULE DESCRIPTIONS

This appendix contains the first-cut module descriptions of the modules shown in Figure 5-2 through 5-6. The module descriptions are detailed and give a good indication of the design of the modules. The 26 module descriptions are:

1 - 1	Process Input
2 - 2.1	Track A/I
3 - 2.2	Convert Coordinates
4 - 2.3.1	Determine Track Type
5 - 2.3.2	Add New Track
6 - 2.3.3	Convert Delete Track
7 - 2.3.4	Update Old Track
8 - 2.3.5	Process Error Messages
9 - 2.4	Track A/I
10 - 2.5	Convert Coordinates
11 - 2.6.1	Determine Type Track
12 - 3	Convert Environmental Data
13 - 4.1	Select Function
14 - 4.2	Display Menu
15 - 4.3	Display Resources
16 - 4.4	Display Engagement Graphics
17 - 4.5	Display Environmental Data
18 - 4.6	Display Ship Parameters
19 - 4.7	Console Display A/I
20 - 4.8	Convert Track to Screen Coordinates
21 - 5.2.1	Optimize Target Track Data
22 - 5.2.2	Calculate Uncertainty Ellipse
23 - 5.2.3	Calculate Acquisition Ellipse
24 - 5.2.4	Check for Optimized Data
25 - 5.4.1	Engagement A/I
26 - 5.4.2	Order Engagement

Module Description

1. Module Name: 1 Process Input

2. Module Purpose: Process operator manual input and act as transform for manual operations.

3. Module Interface Definition

   a. Inputs: Manual track data, environmental data, manual display order, engagement order, menu status.

   b. Outputs: Manual track data, environmental data, manual display order, engagement order.

   c. Miscellaneous (timing, hardware dependence, etc): Correct menu information determines which transaction will occur.

4. Module Processing Narrative Description: Process manual input module takes all user input, determines the type of the input data by checking the current display menu, from which the desired input can be determined. After state determination, the input type is determined and the proper choice of transaction is made. The data input is then sent down the specific transaction route.

5. Superordinate Modules: None.

6. Subordinate Modules: All the update track data base modules, all the convert environmental modules, all the decode output modules, all the plan engagement modules.

7. Error Detection/Handling: Formatted messages are displayed on the display noting an error for: invalid input, out-of-range input.

8. Design Decisions:

   a. This design relies heavily on the details of the plasma display unit which is still in the development stages through McDonnell Douglas Astronautics. Design decisions concerning error messages are not yet clear.

Module Description

1. Module Name: 2.1 Track A/I

2. Module Purpose: Provide an artificial interface between the Process Manual Input Module and Convert Manual Coordinates.

3. Module Interface Definition

   a. Inputs: Manual Track Data.

   b. Outputs: Manual Track Data.

   c. Miscellaneous (timing, hardware dependence, etc): Dependent on concurrent scheduler. Expect interrupts to inform scheduler when manual input is ready for processing.

4. Module Processing Narrative Description: Track A/I is an abstract interface module to provide isolation to the track routines so that if the input to the Update Track data base modules is changed due to redesign, only this Track A/I module will require redesign.

5. Superordinate Modules: 1 Process Module.

6. Subordinate Modules: 2.1 Convert Manual track Coordinates and those subordinate to it.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Must match input data to required output for the subordinate modules in Update Track Data Base.

Module Description

1. Module Name: 2.2 Convert Coordinates

2. Module Purpose: Convert display coordinates to proper track data base coordinates.

3. Module Interface Definition

   a. Inputs: Manual track data.

   b. Outputs: Converted manual track data.

   c. Miscellaneous (timing,hardware dependence, etc): None.

4. Module Processing Narative Description: Convert coordinates takes the display coordinates and converts to the track data base requirements.

5. Superordinate Modules: 2.1 Track A/I

6. Subordinate Modules: 2.2.1 Determine Type Track and Subordinate.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Must decide the coordinate system which will drive the track data base and this module design.

Module Description

1. Module Name: 2.3.1 - Determine Track Type.

2. Module Purpose: Determine if new, old or delete track.

3. Module Interface Definition

    a. Inputs: Track Data, clock.

    b. Outputs: New track data, old track data, delete track data.

    c. Miscellaneous (timing, hardware dependence, etc): None

4. Module Processing Narrative Description: Determine Track Data reviews the manual track data and determines what kind of a track it is. This information to determine the type track data must be contained in the data base. Time stamps new tracks.

5. Superordinate Modules: 2.2 Convert Coordinates.

6. Subordinate Modules: 2.3.3 Convert Delete Track, 2.3.4 Update Old Track, 2.3.4 Add New Track and Subordinates.

7. Error Detection/Handling: None.

8. Design Decisions:

    a. data base must hold a key as to the type track in data base

Module Description

1. Module Name: 2.3.2 - Add New Track.

2. Module Purpose: Add a new track (manual or NTDS) to the data base.

3. Module Interface Definition

   a. Inputs: NTDS and/or Manual new track data.

   b. Outputs: New track to data base.

   c. Miscellaneous (timing, hardware dependence, etc): None.

4. Module Processing Narrative Description: Add New Track adds a new track to the track data base in the proper data base format.

5. Superordinate Modules: 2.6.1 and 2.3.1 Determine Type Track.

6. Subordinate Modules: None.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Insure new track added in proper data base format.

Module Description

1. Module Name: 2.3.3- Convert Delete Track.

2. Module Purpose: To place track number of a manual or NTDS track to be deleted.

3. Module Interface Definition

   a. Inputs: NTDS or Manual track data.

   b. Outputs: Track number of track to be deleted to Delete Track data base.

   c. Miscellaneous (timing, hardware dependence, etc): None

4. Module Processing Narrative Description: Convert Delete Track takes the track number from the NTDS or manual track data it receives as input and copies the track number of this track to be deleted so that the NTDS and Manual Determine Type Track Modules can check the deleted track files and thereby not inadvertantly add a deleted track to the data base.

5. Superordinate Modules: 2.3.1 and 2.6.1 Determine Type Track Modules.

6. Subordinate Modules: None.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Data base requirement is to retain only the track number, make module function simple, copy track number into delete track data base.

Module Description

1. Module Name: 2.3.4 - Update Old Track.

2. Module Purpose: Update an old track in the existing track data base.

3. Module Interface Definition

   a. Inputs: Old NTDS or manual track, track from track data base.

   b. Outputs: Error message to display, update of track data base.

   c. Miscellaneous (timing, hardware dependence, etc): None.

4. Module Processing Narrative Description: Update Old Track takes either input from Manual or NTDS to update the existing track data base. Also must delete tracks when ordered. If there is no track with that track number, then an error message is sent to the display, and the next track is accepted.

5. Superordinate Modules: 2.3.1 and 2.6.1 Determine Type Track Module.

6. Subordinate Modules: 2.3.5 Process Error Messages

7. Error Detection/Handling: Error message sent to the display.

8. Design Decisions:

   a. Make as simple as possible by only requiring check of track numbers to speed update, or error message.

Module Description

1. Module Name: 2.3.5 Process Error Messages

2. Module Purpose: If error notification is receieved process proper error messages.

3. Module Interface Definition

   a. Inputs: Error motification.

   b. Outputs: Display order.

   c. Miscellaneous (timing, hardware dependence, etc): None.

4. Module Processing Narative Description: Process of appro- appropriate error message is to be determined.

5. Superordinate Modules: 2.3.4 Update Old Track.

6. Subordinate Modules: None.

7. Error Detection/Handling: This is the error message handler.

8. Design Decisions:

   a. Must determine the type of error messages desired.

Module Description

1. Module Name: 2.4 Track A/I

2. Module Purpose: Provide an artificial interface between the NTDS Interface and Convert Coordinates.

3. Module Interface Definition

   a. Inputs: NTDS all track data.

   b. Outputs: NTDS all track data.

   c. Miscellaneous (timing, hardware dependence, etc): Dependent on concurrent scheduler. Expect interrupts to inform scheduler when NTDS input is ready for processing.

4. Module Processing Narrative Description: NTDS Track A/I is an abstract interface module to provide isolation to the track routines so that if the input to the Update Track data base modules is changed due to redesign, only this Track A/I module will require redesign.

5. Superordinate Modules: NTDS interface, not part of this system design.

6. Subordinate Modules: 2.5 Convert NTDS Track Coordinates and those subordinate to it.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Must match input data to required output for the subordinate modules in Update Track Data Base.

Module Description

1. Module Name: 2.5 Convert Coordinates

2. Module Purpose: Convert NTDS coordinates to proper track data base coordinates.

3. Module Interface Definition

   a. Inputs: NTDS track data.

   b. Outputs: Converted NTDS track data.

   c. Miscellaneous (timing, hardware dependence, etc): None.

4. Module Processing Narrative Description: Convert coordinates takes the NTDS coordinates and converts to the track data base requirements.

5. Superordinate Modules: 2.4 Track A/I

6. Subordinate Modules: 2.6.1 Determine Track Type.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Must decide the coordinate system which will drive the track data base and this module design.

Module Description

1. Module Name: 2.6.1 Determine Type Track.

2. Module Purpose: Determine if new, old or delete track. Determine if air, surface or subsurface track.

3. Module Interface Definition

   a. Inputs: NTDS track and delete track data.

   b. Outputs: Delete track data, old NTDS track data, new NTDS track data.

   c. Miscellaneous (timing,hardware dependence, etc): None.

4. Module Processing Narative Description: Determine type track reviews the NTDS track data and determines what kind of track it is. This information to determine the type of track data must be contained in the NTDS track data base.

5. Superordinate Modules: 2.5 Convert Coordinates.

6. Subordinate Modules: 2.3.4 Update Old Track
   2.3.2 Add New Track.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Data base must hold a key as to type track in data base.

Module Description

1. Module Name: 3 - Convert Environment Data.

2. Module Purpose: To convert environmental data into required data base format.

3. Module Interface Definition

   a. Inputs: Clock, Environmental data.

   b. Outputs: Update environmental data.

   c. Miscellaneous (timing, hardware dependence, etc): None.

4. Module Processing Narrative Description: Convert Environment Data places the environmental data input of sea state, rain state, and wind state into the environmental data base.

5. Superordinate Modules: 1 - Process Input Module.

6. Subordinate Modules: None.

7. Error Detection/Handling: If data not within prescribed limits, error message is displayed for the operator.

8. Design Decisions:

   a. The requirement to have a data base for environmental data is for neatness and consistency of design.

Module Description

1. Module Name: 4.1 - Select Function.

2. Module Purpose: Determination what to display on the from user driven display changes or from automatic software driven display functions.

3. Module Interface Definition

   a. Inputs: Manual display order, menu status.

   b. Outputs: Any one of the following transactions:
      - order to display menu.
      - order to display resources.
      - order to display engagement.
      - order to display environmental data.
      - order to display ship parameters.

   c. Miscellaneous (timing,hardware dependence, etc):

4. Module Processing Narrative Description: Select Display Function takes a manual order and relays that order to the proper display module. The output is in a transaction format.

5. Superordinate Modules: 1 - Process Input Module.

6. Subordinate Modules: Display menu,resources, engagement graphics, environmental and ship parameter data.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. There must be a means to control the display functions.

Module Description

1. Module Name: 4.2 - Display Menu.

2. Module Purpose: Display the proper menu when correct manual input received.

3. Module Interface Definition

   a. Inputs: Order to display proper menu.

   b. Outputs: Proper display menu.

   c. Miscellaneous (timing, hardware dependence, etc): None.

4. Module Processing Narrative Description: Menus vary depending on the type of operation the operator desires to perform. The operator chooses which menu is desired by manual input, then the order to this module sends the proper menu to the display.

5. Superordinate Modules: 4.1 Select Display Function.

6. Subordinate Modules: 4.7 Console Display A/I.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Display is the key to knowing what the manual input is, the display lets the operator know where he is in the program.

Module Description

1. Module Name: 4.3 - Display Resources.

2. Module Purpose: Display the missile resources.

3. Module Interface Definition

   a. Inputs: Order to display the missile resources.

   b. Outputs: Missile resource display data.

   c. Miscellaneous (timing, hardware dependence, etc): None.

4. Module Processing Narrative Description: The resources of HARPOON missiles including type, location, (other information if desired) is made available for display when ordered.

5. Superordinate Modules: 4.1 Select Display Function.

6. Subordinate Modules: 4.7 Console Display A/I.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Resources are important for operator information, the availability of missile status must be available for display.

Module Description

1. Module Name: 4.4 - Display Engagement Graphics.

2. Module Purpose: Display engagement graphics, e.g. flight path, ellipse of uncertainty, ellipse of acquisition, way point information.

3. Module Interface Definition

   a. Inputs: Display order.

   b. Outputs: Engagement graphics to Display Console.

   c. Miscellaneous (timing, hardware dependence, etc): None.

4. Module Processing Narrative Description: The engagement is the purpose for the HSCLCS, so the graphics associated with these design decisions are imperative to afford the operator a true picture of the engagement.

5. Superordinate Modules: 4.1 Select Display Function.

6. Subordinate Modules: 4.7 Console.Display A/I.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Requires use of many primitive calls to the display CPU chip.

Module Description

1. Module Name: 4.5 - Display Environmental Data.

2. Module Purpose: Display environmental data sends environmental information to the display console.

3. Module Interface Definition

   a. Inputs: Display order.

   b. Outputs: Environmental information to Display Console.

   c. Miscellaneous (timing, hardware dependence, etc): None.

4. Module Processing Narrative Description: Environmental data is important to the engagement calculation, and therefore the operator must be able to see what environmental factors are being taken into consideration.

5. Superordinate Modules: 4.1 Select Display Function.

6. Subordinate Modules: 4.7 Console Display A/I.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Requires display of environmental data when desired.

Module Description

1. Module Name: 4.6 - Display Ship Parameter.

2. Module Purpose: Display ship parameters sends ship parameter information to the display console.

3. Module Interface Definition

   a. Inputs: Display order.

   b. Outputs: Ship parameter data to Display Console.

   c. Miscellaneous (timing, hardware dependence, etc): None.

4. Module Processing Narrative Description: Ship parameter data is important to the engagement calculation, and therefore the operator must monitor the ship parameter factors used for engagement calculations.

5. Superordinate Modules: 4.1 Select Display Function.

6. Subordinate Modules: 4.7 Console Display A/I.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Requires display of ship parameter data when desired.

Module Description

1. **Module Name**: 4.7 - Console Display A/I

2. **Module Purpose**: Artificial interface to display.

3. **Module Interface Definition**

   a. **Inputs**: Variety of display orders.

   b. **Outputs**: Display data to display computer.

   c. **Miscellaneous** (timing, hardware dependence, etc): None.

4. **Module Processing Narrative Description**:

5. **Superordinate Modules**: Modules 4.2, 4.3, 4.4, 4.5, 4.6 and 4.8

6. **Subordinate Modules**: Display screen.

7. **Error Detection/Handling**: None.

8. **Design Decisions**:

   a. Most interface between DPC computer and display computer.

Module Description

1. Module Name: 4.8 Convert Track to Screen Coordinates.

2. Module Purpose: Convert the track data to the screen coordinate system for display.

3. Module Interface Definition

   a. Inputs: Track data from track data base.

   b. Outputs: Convered track data.

   c. Miscellaneous (timing,hardware dependence, etc): Requirement to be determined.

4. Module Processing Narative Description: Convert track to screen coordinates module converts the track data base coordinates to the screen coordinates required.

5. Superordinate Modules: None.

6. Subordinate Modules: 4.7 Console Display A/I

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Must integrate the display screen coordinate system with the requirements of the track data base stored coordinate system.

Module Description

1. Module Name: 5.2.1 - Optimize Target Track Data.

2. Module Purpose: To provide track information to the target optimization engagement routines in subordinate modules so that in the CPU's free time optimum engagements may be calculated.

3. Module Interface Definition

   a. Inputs: Track data, ship parameter data, engagement solution request, order for next track to optimize.

   b. Outputs: Track data.

   c. Miscellaneous (timing, hardware dependence, etc): None.

4. Module Processing Narrative Description: A request by the optimization engagement routines for a track to optimize is filled by providing the tracks within a geographic sector to be determined by the full algorithm. In the case where the operator does not like the engagement order, a request received by this module will take precedence to calculate the desired track the operator wants the missile to take.

5. Superordinate Modules: 5.4.2 Order engagement.

6. Subordinate Modules: 5.2.2 One subordinate module of the optimum engagement calculation routine.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. This module and its subordinates are the most important module in the engagement. They provide an optimum engagement or an engagement requested by the user. This is critical to providing the most efficient use of HARPOON assets.

Module Description

1. Module Name: 5.2.2 - Calculate Uncertainty Ellipse.

2. Module Purpose: Calculates the uncertainty ellipse.

3. Module Interface Definition

   a. Inputs: Track data.

   b. Outputs: Track data and uncertainty ellipse data.

   c. Miscellaneous (timing, hardware dependence, etc): None.

4. Module Processing Narrative Description: This module calculates the ellipse of uncertainty for engagement decision making.

5. Superordinate Modules: 5.2.1 Optimize Track Data.

6. Subordinate Modules: 5.2.3 Calculate Acquisition Ellipse and others in this optimum engagement section.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Required for optimization technique.

Module Description

1. Module Name: 5.2.3 - Calculate Acquisition Ellipse.

2. Module Purpose: To calculate acquisition ellipse.

3. Module Interface Definition

   a. Inputs: Track data and uncertainty ellipse data.

   b. Outputs: Track data, uncertainty ellipse data, acquisition ellipse data.

   c. Miscellaneous (timing, hardware dependence, etc): None.

4. Module Processing Narrative Description: Calculates the uncertainty ellipse.

5. Superordinate Modules: 5.2.2 Calculate Uncertainty Module.

6. Subordinate Modules: 5.2 Check for Optimized Data.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Required for optimizing engagements.

Module Description

1. Module Name: 5.2.4 - Check for Optimized Data.

2. Module Purpose: This routine is not defined yet. There must be a way to determine when the optimization is reached.

3. Module Interface Definition

   a. Inputs: Track data, uncertainty and acquisition ellipses, time.

   b. Outputs: Update engagement track data base and look for another optimization of the engagement desired.

   c. Miscellaneous (timing,hardware dependence, etc): None.

4. Module Processing Narrative Description: Unknown.

5. Superordinate Modules: 5.2.3 Calculate Acquisition Ellipse.

6. Subordinate Modules: None.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. required for automatic optimization to work.

Module Description

1. Module Name: 5.4.1 - Engagement A/I.

2. Module Purpose: To provide an artificial interface between Process Input and Order Engagement Modules.

3. Module Interface Definition

   a. Inputs: Engagement order.

   b. Outputs: Engagement order.

   c. Miscellaneous (timing, hardware dependence, etc): None.

4. Module Processing Narrative Description: Provides artificial interface.

5. Superordinate Modules: 1 Process Input Module.

6. Subordinate Modules: 5.4.2 Order Engagement Module.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Provide ease in changing hardware inputs, or software inputs, so only this artificial interface is changed.

Module Description

1. Module Name: 5.4.2 - Order Engagement.

2. Module Purpose: relay orders to the launcher and missile concerning missile to be launched, and the missile parameters required to achieve a successful launch.

3. Module Interface Definition

   a. Inputs: Launcher and missile status, ship parameter data, engagement track data (optimized engagement solution), engagement order.

   b. Outputs: Launcher missile status, engagement solution request.

   c. Miscellaneous (timing, hardware dependence, etc): None.

4. Module Processing Narrative Description: Processes operators request to engage a track. Allows operator to accept optimized solution or pick a launch path as he desires. Sends orders to launcher and missile concerning the decision to fire.

5. Superordinate Modules: 5.4.1 Engagement A/I.

6. Subordinate Modules: 5.2.1 Optimize Target Track Data, and Launcher and missile thru LRU.

7. Error Detection/Handling: None.

8. Design Decisions:

   a. Most important module, required to fire the weapon from this console.

LIST OF REFERENCES

1. Pressman, Roger S., Software Engineering: A Practitioner's Approach, Mcgraw-Hill Book Company, 1982.

2. Weinberg, Victor, Structured Analysis, Yourdon,Inc., 1979.

3. Jensen, R. W. and Tonies, C. C., Software Engineering, Prentice Hall, Inc., 1979.

4. Commander, Naval Sea Systems Command, SW261-23-MMO-010, CG-47 Class HARPOON Weapon System, Description Operation, and Maintenance, 1 January 1982 (Preliminary Copy).

5. Commander, Naval Sea System Command, Draft Specification for Performance and Design Requirements for the HARPOON Ship Command and Launch Control Set AN/SWG-1A(V) (HSCLCS) 24 September 1982.

BIBLIOGRAPHY

Artzer, S. P. and Neidrauer, R. A., Software Engineering: A Primer for the Project Manager, Master's Thesis, Naval Postgraduate School, Monterey, California, June 1982.

Calingert, Peter, Operating System Elements a User Perspective, Prentice-Hall, Inc., N.J., 1982.

Dijkstra, E. W., Cooperating Sequential Processes in Programming Languages, Academic Press, N.Y., 1968.

Hoare, C. A. R., "Monitors: An Operating System Structuring Concept", Communications of the ACM, 17, 10(Oct 1974), 549-557.

Reed, D. P., Kanodia, R. K., "Synchronization with Eventcounts and Sequencers, Communications of the ACM, 22, 2(Feb. 1979).

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 52 Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
4.	LCDR O. L. Sentman, USN Strategic Systems Project Office Navigation Branch (SP-24135) Department of the Navy Washington, D.C. 20376	1
5.	LT Randall K. Maroney, USN NAVPRO, General Dynamics Phalanx Pamona, CA 91766	1
6.	NSWSES Code 4613 (Mr. Paul Gognon) Port Heuneme, CA 93043	1
7.	Naval Sea Systems Command Department of the Navy ATTN: Mr. Lee Minin NAVSEA Code SEA-62WB Washington, D.C. 20362	1
8.	Mr. Fred S. Gais Director - HARPOON Ship Integration McDonnell Douglas Astronautics Company St. Louis Division P.O. Box 516 St. Louis, MO 63166	1

9. LCDR Ronald Modes, USN, Code 52MF 4
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93940

10. LCDR William Shockley, USN, Code 52SK 1
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93940

11. Associate Professor Uno R. Kodres, Code 52Kr 1
Department of Computer Science
Naval Postgraduate School
Monterey, California 93940